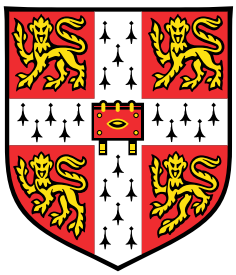
Numerical stability of Monte Carlo neutron transport and isotopic depletion for nuclear reactor analysis



Paul Cosgrove

Supervisor: Dr. E. Shwageraus

Advisor: Dr. G.T. Parks

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Pembroke College October 2020

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Paul Cosgrove
October 2020

Acknowledgements

First, I'm indebted to the Nuclear Energy MPhil course and its management during the academic year of 2015-2016 for inducting me into nuclear – I would not trade the last few years for anything, and I hope to continue learning about nuclear energy for much longer.

Thanks to my supervisor, Dr. Eugene Shwageraus, for his patience, wisdom, and encouragement throughout the last five years, and for twice accepting me as a student. He has been an inspiration in nuclear engineering and beyond, and – based on the testaments of many students elsewhere – I am extremely lucky to have had a supervisor with whom I have so enjoyed working.

Thanks also to Dr. Geoff Parks for his efficiency and availability when any problems arise, for his super-human spell-checking, for greatly assisting in obtaining funding, and for dragging my progressive rock tastes into the modern era.

Many thanks to Jo Boyle who has certainly done more for me than I know by navigating layers of administration, ensuring the smooth running of the Nuclear Energy group, and vigilantly guarding the office coffee supply and the sanctity of Treat Tuesday, without which this thesis would not have been possible.

I am grateful to Dr. Dan Kotlyar for his patient assistance during the early stages of the PhD, allowing me to begin this work in earnest. I have also greatly benefited from discussions with him and many other nuclear engineers – such as the Serpent team and Dan's students at Georgia Tech – on many occasions since.

Thanks to the EPSRC ICO-CDT for their generous funding and for enabling the conference trips and the nuclear tourism without which this PhD would not have been quite the same – even at the price of occasional visits to Birmingham.

Throughout the PhD, I have often had occasion to reflect on how fortunate I am to be a part of such a warm and sociable research group. Much of the fun and enthusiasm for my work has been thanks to the other characters that have inhabited BS3-05 over the years. Between arguing beside the coffee machine, Thirsty Thursdays, and Reading Club ranting, it's been a blast. Special thanks are due to my contemporaries, Mikolaj Kowalski and Nathaniel Read, both of whom have taught me so much and helped me to be a better researcher.

I am also very grateful to Pembroke College for giving me food, shelter, and entertainment for the last four years, and for all of the friends I have made there. There are far too many of them to thank individually, but a special mention must go to Chris Howland and Ben Jackson for their relevance to this thesis in providing useful literature and helping to elevate the mathematical understanding of a mere engineer.

Although these occasions are long over now, I must thank my sister, Claire, for the innumerable breakfasts, dinners, brunches, and cups of tea she treated me to at Corpus Christi while she was still there. On that note, thanks to The Trio for always keeping me in craic.

Finally, thanks to my parents who have always provided me with every resource I could possibly require to succeed.

Abstract

Coupling Monte Carlo neutron transport with isotopic depletion is known to produce non-physical results for large reactor geometries. This thesis begins with a survey of the stable methods used to avoid this and highlights some of their drawbacks.

Chapter 2 introduces the known phenomenon of neutron clustering in Monte Carlo as a factor strongly contributing to previous reports of instability. It is shown that attempting to minimise clustering effects produces more stable burn-up calculations. Furthermore, accounting for clustering is shown to allow for the accurate simulation of xenon transients in simple systems which have been noted to pose a challenge for burn-up simulations. Finally, it is demonstrated that neutron clustering can also affect burn-up simulations substantially even when xenon equilibrium is enforced, namely by way of the previously hypothesised 'gadolinium instabilities'.

Chapter 3 begins by highlighting how implicit burn-up schemes may be viewed as root-finding schemes for a discrete map. It is shown that, for reasonably long time-steps, the corrector step of predictor-corrector schemes does not succeed in locating the root (or the stable solution) of this map. Hence, relaxation schemes are introduced; relaxation schemes have been applied to neutronics/depletion coupling previously in the form of the stochastic approximation, although this is relatively inefficient. The relaxation scheme proposed here uses a fixed relaxation factor (rather than the variable factor previously used) and demonstrates its improved stability and computational efficiency compared to the stochastic approximation. The same investigations are applied to a depletion problem where equilibrium xenon is enforced – it is seen that this, too, can be unstable, but is resolvable through relaxation.

Chapter 4 performs a Von Neumann stability analysis of a simple coupled neutron diffusion-depletion system. Extending a previous analysis, this chapter provides a justification for applying a relaxation to predictor-corrector schemes, shows the possibility of obtaining symmetric burn-up instabilities, and demonstrates that no neutronics-depletion coupling scheme, without relaxation, is assuredly more stable than another, depending on the depletion system in question.

Finally, Chapter 5 summarises the findings, proposes an explanation regarding general cases of burn-up instability, and suggests future work.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Nuclear reactor simulation

Nuclear reactors are inherently 'multi-physical' systems, i.e., to accurately predict their behaviour, a number of interacting physical phenomena must be considered simultaneously. For example, the power distribution in a reactor depends on the relative rates of nuclear fission across the system, as determined by neutron transport. However, fission produces heat, reducing the local density of coolant, decreasing the degree of neutron moderation, and thus driving a decrease in the local fission rate. Similarly, a high local fission rate at a point in time will produce neutron-absorbing fission products at a higher rate, eventually causing a local depression in the fission rate at some future time. The coupling of neutronics with thermal-hydraulics, and neutronics with isotopic depletion are two of the more obviously intertwined sets of phenomena present in reactors. Other phenomena also play significant roles in reactor operation, e.g., coolant chemistry, fuel mechanics, fission gas behaviour.

Qualitatively, how a 'multi-physics' system will behave is not necessarily difficult to imagine. However, a quantitative analysis is quite another task: resolving different physical regimes simultaneously often amounts to obtaining the solution of a non-linear system of partial differential equations. This is a difficult task for prospective reactor designers, operators, and regulators, necessitating the use of a computer. Approximations can simplify the required effort significantly but may not always be acceptable; for example, a liquid-fuel reactor may require a more thorough analysis of the fluid mechanics and chemistry than a traditional Pressurised Water Reactor (PWR) to satisfy a regulator. Alternatively, when attempting to up-rate the power of a reactor, a safety case must also be made, demonstrating with high certainty that thermal limits and other safety requirements are not breached. Hence, in lieu of conducting many expensive experiments, there is a need for high-fidelity multi-

physics simulation capabilities and, for the sake of the designer, operator, or regulator in question, these capabilities should be relatively fast, cheap, and reliable.

1.2 Neutron transport

Nuclear reactors exploit the fissioning of atoms to produce energy: when an atom of, say, uranium-235 is split by a neutron, it produces approximately 200 MeV of energy and, depending on the instigating neutron's energy, two or more daughter neutrons. The production of these daughter neutrons allows for the fissioning of more atoms and so the possibility of a chain reaction. To consider neutron transport or neutronics is to consider the behaviour of an ensemble of neutrons, often with the intent of establishing the state of the chain reaction and obtaining a space and/or time distribution of nuclear reaction rates.

Before describing the governing equation, it is worth defining some of the important quantities for the sake of clarity. Naturally, neutron transport is concerned with the space- and time-dependent neutron density, denoted $n(\boldsymbol{r},t)$, at position $\boldsymbol{r}$, at time t. Nuclear reaction rates at a given time and place are not only a function of the density of neutrons, but also their energy, E. Part of this energy dependence is due to the nuclear structure of the matter with which neutrons interact.

Typically, at low neutron energies (on the order of 1 eV) nuclear reactions are much more frequent than at the high energies at which neutrons are born (about 1 MeV). Qualitatively, this can be understood by less energetic, slower neutrons spending more time in the vicinity of the nuclei with which they might interact, as compared to fast neutrons. However, in intermediate energy regions, the propensity for nuclear reactions often varies by many order of magnitude with even small changes in a neutron's energy. This 'resonance region' occurs as a result of the combined energy of the neutron and nucleus corresponding to an energy level of the resulting compound nucleus. For many nuclei, the often complicated energy dependence of nuclear reactions is tabulated as a 'microscopic cross-section', denoted by $\sigma_{\mathrm{r}}(E)$ for a particular reaction mode r, with the usual unit of barns where $1\mathrm{b} = 10^{-24}\mathrm{cm}^{-2}$. For nuclear reactors, the most important reaction modes are fission, capture, and both elastic and inelastic scattering. The fission and capture cross-sections of U-238 are shown in Fig. 1.1.

Hence, a macroscopic nuclear reaction rate is an integration over some volume of space and the energy of the colliding neutrons. For reaction mode r, the total reaction rate over a volume V at time t is:

$$R_{\mathrm{r}}(t) = \int_0^\infty \mathrm{d}E \int_V \mathrm{d}V \sigma_{\mathrm{r}}(E) N(\boldsymbol{r}) v(E) n(\boldsymbol{r},t) = \int_0^\infty \mathrm{d}E \int_V \mathrm{d}V \Sigma_{\mathrm{r}}(\boldsymbol{r},E) \phi(\boldsymbol{r},E,t) \qquad (1.1)$$

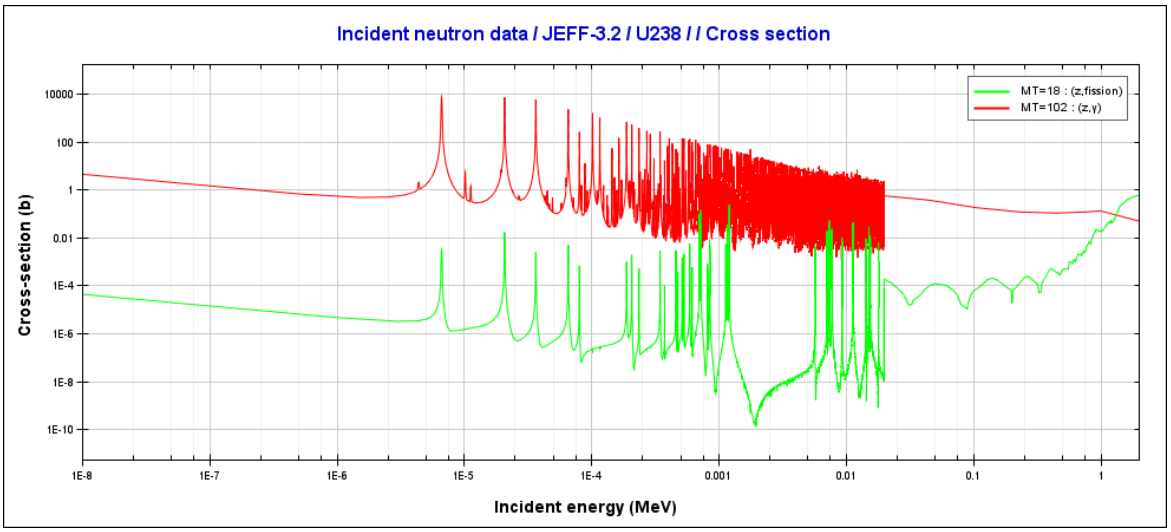


Fig. 1.1 The fission and capture cross-sections of U-238, plotted by JANIS [91].

where $v(E)$ is the neutron speed and Σ_r is the macroscopic cross-section, defined as:

$$\Sigma_\mathrm{r} = \sigma_\mathrm{r} N \tag{1.2}$$

or, where more than one species of nuclide is present:

$$\Sigma_\mathrm{r} = \sum_{i=1}^{I} \sigma_{\mathrm{r},i} N_i \tag{1.3}$$

and ϕ is known as the scalar neutron flux, defined as:

$$\phi = vn \tag{1.4}$$

Another useful quantity is the angular neutron flux, $\psi(\boldsymbol{r},\boldsymbol{\Omega},E,t)$, where $\boldsymbol{\Omega}$ is the neutron direction of flight. The scalar flux is obtained from the angular flux as:

$$\phi(\boldsymbol{r},E,t) = \int_{4\pi} \mathrm{d}\boldsymbol{\Omega}\psi(\boldsymbol{r},\boldsymbol{\Omega},E,t) \tag{1.5}$$

In a nuclear reactor (as opposed to a neutron star, for example), for a static nuclide density field, neutron transport is governed by what is termed the linear, time-dependent neutron

transport equation:

$$\begin{aligned}
&\frac{1}{v(E)}\frac{\partial \psi}{\partial t} + \boldsymbol{\Omega}\cdot\nabla\psi(\boldsymbol{r},\boldsymbol{\Omega},E,t) + \Sigma_{\mathrm{t}}(\boldsymbol{r},E)\psi(\boldsymbol{r},\boldsymbol{\Omega},E,t) \\
&= (1-\beta(\boldsymbol{r}))\frac{\chi_{\mathrm{p}}(\boldsymbol{r},E)}{4\pi}\int_0^\infty \mathrm{d}E'\bar{\nu}\Sigma_{\mathrm{f}}(\boldsymbol{r},E')\int_{4\pi}\mathrm{d}\boldsymbol{\Omega}'\psi(\boldsymbol{r},\boldsymbol{\Omega}',E',t) \\
&\quad + \int_0^\infty \mathrm{d}E'\int_{4\pi}\mathrm{d}\boldsymbol{\Omega}'\Sigma_{\mathrm{s}}(\boldsymbol{r},\boldsymbol{\Omega}'\to\boldsymbol{\Omega},E'\to E)\psi(\boldsymbol{r},\boldsymbol{\Omega}',E',t) \\
&\quad + \sum_{i=1}^{N}\frac{\chi_{\mathrm{d},i}(E)}{4\pi}\lambda_i C_i(\boldsymbol{r},t) + S(\boldsymbol{r},\boldsymbol{\Omega},E,t)
\end{aligned} \tag{1.6}$$

This equation describes the balance of neutrons in position, angle, energy and time. The left-hand side contains terms describing, respectively, the rate of change of neutron density, the loss of neutrons through streaming, and the loss of neutrons through all reactions with matter – Σ_{t} is the total cross-section, the sum of all reaction cross-sections.

The first term on the right-hand side describes the gain of prompt neutrons from fission reactions: $1-\beta$ is the fraction of neutrons which are born directly from fission (or are prompt), χ_{p} is the spectrum of fission neutrons, $\bar{\nu}\Sigma_{\mathrm{f}}$ is the product of the average number of neutrons produced per fission and the macroscopic fission cross-section. Next is the production of neutrons from scattering reactions, where $\Sigma_{\mathrm{s}}(\boldsymbol{r},\boldsymbol{\Omega}'\to\boldsymbol{\Omega},E'\to E)$ is the macroscopic neutron scattering kernel which combines both the macroscopic scattering cross-section and the angular distribution of scattered neutrons. The third term describes the production of delayed neutrons from fission products or 'precursors'. Delayed neutrons are born from the decay of these precursors, with density C, as determined by a decay constant λ, and with an energy spectrum $\chi_{\mathrm{d}}(E)$. Delayed neutrons and their precursors are often lumped into a number of discrete groups, N, typically 6, here identified by the index i. The precursor density is determined by the local fission and decay rate, with a separate equation for the evolution of each precursor group. Finally, the term $S(\boldsymbol{r},\boldsymbol{\Omega},E,t)$ describes an external source of neutrons.

Traditionally, reactor kinetics are considered separately from neutron transport, allowing the time-dependence of Eq. (1.6) to be neglected. Given nuclear reactors are nominally operating at steady-state or under very slow transients, this is justifiable. Commonly in reactor analysis, external neutron sources are also assumed negligible. Hence, the steady-

state neutron transport equation is:

$$\begin{aligned} &\boldsymbol{\Omega}\cdot\nabla\psi(\boldsymbol{r},\boldsymbol{\Omega},E)+\Sigma_{\mathrm{t}}(\boldsymbol{r},E)\psi(\boldsymbol{r},\boldsymbol{\Omega},E)\\ &=\frac{\chi_{\mathrm{eff}}(\boldsymbol{r},E)}{4\pi k_{\mathrm{eff}}}\int_0^\infty \mathrm{d}E'\,\bar{\nu}\Sigma_{\mathrm{f}}(\boldsymbol{r},E')\int_{4\pi}\mathrm{d}\boldsymbol{\Omega}'\psi(\boldsymbol{r},\boldsymbol{\Omega}',E')\\ &+\int_0^\infty \mathrm{d}E'\int_{4\pi}\mathrm{d}\boldsymbol{\Omega}'\Sigma_{\mathrm{s}}(\boldsymbol{r},\boldsymbol{\Omega}'\rightarrow\boldsymbol{\Omega},E'\rightarrow E)\psi(\boldsymbol{r},\boldsymbol{\Omega}',E') \end{aligned} \tag{1.7}$$

In removing the time-dependence, two important changes have been made to Eq. (1.6): the first is that, without time-dependence, delayed neutrons can be treated as prompt neutrons but with a different energy distribution. This necessitates a modification of the fission neutron spectrum from $\chi_{\mathrm{p}}(\boldsymbol{r},E)$ to $\chi_{\mathrm{eff}}(\boldsymbol{r},E)$. This combines the pure fission and precursor spectra into one 'effective' spectrum. The second is the appearance of k_{eff}, also known as 'k-effective' or the criticality eigenvalue. k_{eff} occurs when separating neutronics from kinetics as a means of artificially forcing the neutronics solution into a steady-state. In particular, if a reactor is 'super-critical' or has a growing neutron population, the neutron population will grow over time – k_{eff} in this case will be greater than 1, thus depressing the fission neutron production term to enforce the steady state. Conversely, for a 'sub-critical' reactor where the neutron population is decaying, k_{eff} is less than 1, enhancing the fission neutron production. In the case of criticality, about which a reactor normally operates, $k_{\mathrm{eff}}=1$, a true steady-state exists and no modification of the fission source is necessary.

Given its linearity in the neutron flux, Eq. (1.7) can be more compactly stated as a generalised eigenvalue problem:

$$\mathscr{L}\psi=\frac{1}{k_{\mathrm{eff}}}\mathscr{F}\psi \tag{1.8}$$

Here $\mathscr{L}\psi$ lumps together all terms in Eq. (1.7) aside from fission neutron production which is denoted by $\mathscr{F}\psi$. Eq. (1.8) and its approximations have been and continue to be the primary means of studying static reactor core neutronics. However, deterministic solutions of Eq. (1.8) present a number of constraints on the analysis. Namely, given the complicated phase space of a reactor where angular neutron fluxes may vary dramatically in space, angle, and energy, severe discretisation is often required for accuracy. Energy and angle are particularly challenging: due to resonances, accurately resolving the energy variable necessitates an impractically fine energy discretisation if performed naïvely. A common approach to this difficulty is to perform two calculations; the first on a geometrically simpler problem but with a fine energy discretisation and explicit treatment of resonances in order to generate a representative neutron spectrum. With this spectrum, the cross-sections featuring in Eq. (1.7) can be condensed from a fine to a coarse energy discretisation, using a number of approximations. The second calculation uses this coarse but theoretically-equivalent

energy discretisation to solve the full problem under consideration. Treatment of the angular variable depends on the solution method in question, of which there are several. Most prominent are the method of characteristics, discrete ordinates or S_n, the method of spherical harmonics or P_n, and the diffusion approximation. The description of each of these methods is beyond the scope of this thesis, however, the angular treatment of each assumes either that the angular variable can be accurately represented using a number of discrete streaming directions or through a functional expansion of the angular flux in spherical harmonics up to a chosen order. For a detailed exposition on the above methods, see the following references: [1, 6, 8, 23, 25, 73]

Although relatively computationally inexpensive, other practical limitations on deterministic transport methods exist, namely, the finely discretised phase space necessary for large reactor core problems carries a heavy memory footprint, particularly for 3D calculations. While computational power becomes cheaper and the pursuit of higher fidelity continues, alternative approaches to neutronics become increasingly attractive. This has lead to the current prominence of the Monte Carlo method.

1.2.1 Monte Carlo neutronics

The Monte Carlo method is a name ambiguously given to a number of related statistical techniques for numerical analysis. This thesis is concerned with Monte Carlo integration, or, essentially, the application of statistics to calculating complicated integrals. The modern form of Monte Carlo simulation was devised for neutron transport calculations in the design of a thermonuclear weapon by Stanislaw Ulam and John Von Neumann [82] but can equally be applied to finance [14], fluid mechanics [7], chemistry [87], and many other fields.

Applied to neutron transport, the Monte Carlo method involves simulating the lives or 'histories' of many neutrons from their point of birth in fission until their death by absorption or escape from the boundaries of the system [72]. The behaviour of the particles is aggregated to calculate quantities of interest, e.g., reaction rates, or the criticality eigenvalue. This thesis is concerned with criticality simulations which solve an analogue of Eq. (1.8); Monte Carlo can also be used to simulate source driven problems, alternative eigenvalue problems (such as the α-eigenvalue) [101], or even time-dependent system [90]. Although not universal among Monte Carlo codes, most take advantage of exact, tabulated nuclear data: cross-sections and scattering distributions are treated continuously, rather than in discrete form as in deterministic methods – complicated resonance structures in cross-sections can therefore be treated explicitly, without approximation. Similarly, Monte Carlo simulation imposes no limit on the geometric complexity of the systems being analysed – there is no requirement for meshing and all features of the geometry can, in principle, be treated exactly. Consequently,

Monte Carlo simulations are often regarded as a standard for physical fidelity, comparable even with experimental measurements.

The main counterpoint to Monte Carlo's advantages is its computational expense; the extent of this slow-down, relative to a deterministic solver, is problem-dependent, and also dependent upon the simulation settings of the Monte Carlo and deterministic solvers. The computational expense stems, in part, from Monte Carlo's second disadvantage: as Monte Carlo is probabilistic in nature, all results have a degree of statistical uncertainty, determined by the number of particles simulated. In particular, the standard deviation of a Monte Carlo estimator falls as $\frac{1}{\sqrt{n}}$, where n is the number of particles simulated. Hence, to halve the statistical error, one must quadruple the number of particles simulated, quadrupling the expense. Nevertheless, readily available computing power coupled with Monte Carlo's ease of parallelisation has elevated its status from 'method of last resort' to being a ubiquitous reactor physics tool [11]. This is evidenced by the swell of Monte Carlo codes written or drastically re-written within the last decade or so [5, 12, 18, 37, 64, 69, 75, 77, 83, 86, 98, 100]. This surge has coincided with a growth in capabilities and applications: although once used primarily for criticality and shielding, Monte Carlo neutronics solvers now support coupling with depletion [39, 69] and thermal-hydraulics [36], in-line photon transport [37], sensitivity and uncertainty quantification [57, 84], and many other recent developments.

While exceedingly popular in research applications, Monte Carlo simulation remains prohibitively expensive for routine industrial reactor analysis. Rather, the common approach is using a lattice physics tool (e.g., WIMS [71] or CASMO [81]) to generate multi-group cross-sections which are used in a full-core nodal diffusion simulator (e.g., PANTHER [45] or SIMULATE [4]). Presently, it is not inconceivable that Monte Carlo codes might routinely be used for cross-section generation [67, 68], but, for the moment, their use as a full reactor analysis tool is more limited to the likes of one-off benchmarking simulations. Nevertheless, efforts are well under way in certain spheres of nuclear engineering to make Monte Carlo simulations more routine: the MC21 code is designed with the explicit aim of making Monte Carlo a 'mainstream design tool' [37].

As stated above, Monte Carlo proceeds by simulating many neutron histories. However, in criticality calculations, commonly there are three parameters that must be decided by a code user before a calculation can proceed, namely, the population size, the number of active cycles, and the number of inactive cycles. These parameters stem from the algorithm most often implemented in Monte Carlo: the 'method of successive generations' [70]. The method of generations proceeds by simulating a fixed number of particles (the population size) for a specified number of generations, emulating the physical chain reaction that occurs in a multiplying system. Particles produced by fission in one generation are simulated

in the subsequent generation until the simulation is complete, albeit, with some particles either discarded or duplicated to maintain the fixed population size. The total number of generations or 'cycles' is the sum of the number of active cycles and inactive cycles. Monte Carlo criticality simulations also amount to solving the generalised eigenvalue problem in Eq. (1.8), or, more precisely, the following fission source eigenvalue problem:

$$S = \frac{1}{k_{\text{eff}}} \mathscr{M} S \tag{1.9}$$

where S is the fission neutron source distribution and $\mathscr{M}$ is the operator describing the transport and collision processes that fission neutrons undergo. The fission source distribution is unknown initially and must be guessed. However, after a sufficiently large number of generations with a sufficiently large population size, the source should (hopefully) converge to the fundamental mode – this is analogous with power iteration from basic linear algebra. The cycles during which the source is still converging should not have any statistics scored due to the use of a biased fission source – these are the inactive cycles. Of course, to obtain meaningful results, one must run a number of generations during which particle transport contributes to statistical estimators of interest – the active cycles.

A number of best practices [10] and diagnostics [96] exist to ensure that Monte Carlo simulations are sufficiently well converged and their results are not contaminated by the effects of inter-generational correlations between particles or other biases [9]. The nature and effects of correlation are described further in Chapter 2. That being said, when conducting a simulation for the first time, the numbers of particles and cycles to use can only be estimated by heuristics or based on experience. Following from this, source convergence acceleration methods have become popular as a means of reducing the number of inactive cycles necessary in a Monte Carlo simulation; these techniques include fission matrix acceleration [15], the coarse mesh finite difference method [42], and the response matrix method [66], among others.

1.3 Isotopic depletion

The evolution of a material subject to a flux of neutrons is of obvious interest to nuclear engineers: this information determines how long a reactor will operate, whether adjustments must be made to ensure criticality, and the toxicity, radioactivity, and proliferation risk of a reactor's fuel or components on their eventual disposal. This evolution is governed by the rate at which various nuclear reactions occur within a material, the most familiar of these reactions being nuclear decay: unstable isotopes (such as U-235, U-238, Pu-239, Co-60,

to name but a few) will decay spontaneously by α-, β-, or γ-decay, transmuting into a different isotope in the case of α- and β-decay, while releasing another particle in the process. Although the prediction of an individual decay event is impossible, for a large collection of isotopes, the change in the isotopes can be accurately predicted by the equation:

$$\frac{\mathrm{d}N}{\mathrm{d}t} = -\lambda N \tag{1.10}$$

where N is the number or number density of the isotope in question, and λ is the characteristic decay constant of the isotope. That being said, an isotope may decay by multiple mechanisms at different rates, in which case λ is the sum of the decay constants for each mechanism.

In a neutron flux, isotopic evolution is also governed by neutron-induced reactions such as fission, capture, (n,2n) reactions, etc. The rates at which these reactions occur depend on the cross-sections of a material's constituent isotopes and the neutron flux spectrum. This is essentially a repetition of Eq. (1.1). Restating the equation for an infinite, homogeneous medium allows the volume integration to be neglected and the isotope density to be extracted from the integral, giving:

$$\langle \sigma, \phi \rangle = \int_0^\infty \mathrm{d}E \sigma(E)\phi(E) \tag{1.11}$$

Bra-ket notation (familiar from quantum mechanics) has been introduced here to denote an integration over the phase space of interest (or an inner product). Having extracted the nuclide density, the above reaction rate is known as the 'microscopic reaction rate' – the reaction rate per atom of the reacting nuclide. The loss rate of a nuclide with density N by reaction mode r – assuming the flux does not change with time – is then:

$$\frac{\mathrm{d}N}{\mathrm{d}t} = -\langle \sigma_\mathrm{r}, \phi \rangle N \tag{1.12}$$

which is another restatement of Eq. (1.1). An alternative convention is to consider the 'condensed' or 'one-group' microscopic cross-section and a flux magnitude. The former is given by:

$$\bar{\sigma}_\mathrm{r} = \frac{\int_0^\infty \mathrm{d}E \sigma_\mathrm{r}(E)\phi(E)}{\int_0^\infty \mathrm{d}E \phi(E)} \tag{1.13}$$

and the latter by:

$$\bar{\phi} = \int_0^\infty \mathrm{d}E \phi(E) \tag{1.14}$$

which on multiplication reproduce Eq. (1.11).

So far only loss rates have been described. However, each decay, transmutation, or fission will produce additional atoms of a different isotope. α-decay will result in a daughter isotope with a mass number less by 4, and an atomic number less by 2 (as well as atoms of He-4),

while β-decay will result in a daughter with an atomic number incremented either by +1 or -1 (depending on whether it is β^- or β^+). Capture reactions produce a daughter with an atomic mass incremented by +1. Reactions such as (n,2n) or (n,α) can also be accounted for straightforwardly. Fission reactions are more interesting: splitting different isotopes yields different daughter nuclides in different proportions. The amount of a particular daughter produced per fission is known as the fission yield, γ. This fission yield depends on the energy of the neutron inducing fission, but this value is often condensed by the equation:

$$\bar{\gamma} = \frac{\int_0^\infty \mathrm{d}E\,\gamma(E)\sigma_\mathrm{f}(E)\phi(E)}{\int_0^\infty \mathrm{d}E\,\sigma_\mathrm{f}(E)\phi(E)} \tag{1.15}$$

However, it is not only fission which can produce different daughters depending on the neutron spectrum – neutron capture can also result in different daughter isomers. For example, neutron capture by Americium-241 can produce either of the daughters Am-242m or Am-242g, with probability dependent upon the captured neutron's energy. This probability is known as the isomeric branching ratio. Given that these daughters have differing decay chains, presuming a fixed value of these branching ratios can induce errors in systems with different or varying spectra (particularly in actinide isotopics and helium production). Serpent, for example, uses fixed branching ratios by default, corresponding to a PWR spectrum. Where energy-dependent branching ratio data is available, it is possible to calculate the condensed value in a manner analgous to that of fission yields in Eq. (1.15) [40].

Hence, for a material with multiple nuclides, one can write the rate of change of nuclide i due to the reactions of all nuclides j as [6]:

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = \sum_{j=1}^{J} \left[\langle \gamma\sigma_\mathrm{f}^{i,j}, \phi \rangle + \langle \sigma_\mathrm{c}^{i,j}, \phi \rangle + \lambda^{i,j} - \delta_{ij}(\lambda + \langle \sigma_\mathrm{a}, \phi \rangle) \right] N_j \tag{1.16}$$

Here the superscripts i, j above $\gamma\sigma_\mathrm{f}$, σ_c and λ denote the fission yields, reactions or decays which produce nuclide i from nuclide j, while δ_{ij} is the Kronecker delta which demarcates the terms describing nuclide i's loss rate. The branching ratio is implicit in the cross-section $\sigma_\mathrm{c}^{i,j}$. If, rather than referring to N_i and N_j, one refers to a vector of nuclide densities, Eq. (1.16) may be more compactly stated as a matrix differential equation [6]:

$$\frac{\mathrm{d}N}{\mathrm{d}t} = \mathbf{A}N \tag{1.17}$$

where N is now a vector of nuclide densities and $\mathbf{A}$ is known as the burn-up matrix with i, j-th component equal to the term in the square brackets in Eq. (1.16). This equation is known as the Bateman equation and, for a time-independent burn-up matrix, has a formal

solution:

$$N(t+\Delta t) = \exp\left[\mathbf{A}\Delta t\right] N(t) \tag{1.18}$$

where $N(t)$ is the nuclide density vector in a homogeneous region at time t, Δt is the time-step, and $\exp\left[\mathbf{A}\Delta t\right]$ is a matrix exponential. The accurate evaluation of a matrix exponential depends on the exponentiated matrix in question and can be performed by a variety of means [74]. Given the properties of burn-up matrices, the Chebyshev Rational Approximation Method (CRAM) has been shown to be accurate [47, 80]. However, Eq. (1.17) has also been successfully solved by other numerical methods, including more traditional forms of ODE solver [39, 43, 52].

Note, the above discussion applies only to a homogeneous problem: where there are multiple co-existing burnable regions with different compositions or subjected to different neutron fluxes, reaction rates must also be averaged over the volume of each region and calculated separately. Otherwise, the extension is straightforward: nuclide density vectors for separate regions may be concatenated and the burn-up matrix assumes a block diagonal form – essentially each region may be treated entirely separately. That being said, preliminary work has been performed on treating the nuclide spatial distribution continuously rather than imposing a discretisation, although this is presently limited by requiring impractically high-order functional expansions when handling geometries with strong burnable absorbers, such as gadolinium [34].

Performing depletion calculations adds computational expense to a Monte Carlo solver given that many additional integrals equivalent to Eqs. (1.13) must be computed for all isotopes, transmutation and fission reactions, and burnable regions considered — this necessitates multiple cross-section look-up and interpolation operations for every neutron collision in a burnable region. Likewise, flux amplitudes must be obtained for each burnable region using Eq. (1.14), and all fission yields must be computed with Eq. (1.15). One common alternative is to forego scoring many individual cross-sections and instead score a fine-group flux in each burnable region [35, 41]. This minimises computational burden, limiting the look-up operations to determining the energy bin in which to score fluxes and eliminating interpolations. Provided fine-group microscopic cross-sections are available corresponding to the nuclear data used in the transport calculation, the Monte Carlo cross-section condensation, given by Eq. (1.13), can be approximated deterministically as:

$$\bar{\sigma}_{\mathrm{r}} \approx \frac{\sum_g \sigma_{\mathrm{r},g}\phi_g}{\sum_g \phi_g} \tag{1.19}$$

Here g is the index of a particular energy group, ϕ_g is the flux in group g in a given burnable region, and $\sigma_{\mathrm{r},g}$ is the microscopic cross-section of reaction mode r in group g. Errors result

from this discretisation, particularly due to the difficulty of discretising the resonance region – however, these errors are relatively small with a reasonably chosen group structure and can be minimised with a simple treatment to account for discretising resonant cross-sections [35]. This tallying method is used by Serpent [69] and Shift [77].

Discussion so far has also been limited to a constant neutron flux magnitude with a fixed spectrum. Where these vary in time, as in a nuclear reactor, the depletion solver must repeatedly exchange information with a transport solver, with more sophistication necessary to obtain an accurate nuclide evolution.

1.4 Coupled Monte Carlo neutronics and depletion

The neutron transport equation, as shown in Eq. (1.8), has been written without highlighting the inputs necessary to solve a neutronics problem: the geometry, the nuclide density field and the temperature field. Including the nuclide density field (and, implicitly, the geometry), the equation becomes:

$$\mathscr{L}(N)\psi = \frac{1}{k_{\text{eff}}}\mathscr{F}(N)\psi \tag{1.20}$$

Likewise, the Bateman equation, Eq. (1.17), assumed a constant neutron flux. Accounting for the dependence upon the flux gives:

$$\frac{\mathrm{d}N}{\mathrm{d}t} = \mathbf{A}(\psi)N \tag{1.21}$$

Coupled neutronics/depletion problems are governed by this non-linear system of equations and typically assume that the timescale of nuclide evolution is extremely long in comparison to timescales of concern in dynamic neutron transport – hence the use of the eigenvalue form of the transport equation. These equations are also supplemented with boundary conditions and a flux normalisation condition, given the transport equation is an eigenvalue problem. The boundary conditions specify an initial nuclide density field and the treatment of neutrons which reach the system boundary. The normalisation is typically decided by specifying a fission power or powesityr density for the system. The non-linear system could be further extended to couple thermal-hydraulics by including temperature dependence in Eq. (1.20) and introducing the Navier-Stokes and heat conduction equations, with the power distribution obtained from Eq. (1.20) as a heat source.

Physically, the difficulty in solving these coupled equations arises from the amplitude and spectrum of the flux changing while a reactor is depleting – fuel is consumed and absorbing fission products are generated, causing a flux depression in regions where the power may

once have been high, increasing it where power was once low. Therefore, to accurately capture these changes in spectrum and their effects on the nuclide field, a sufficiently fine time discretisation must be used, supplemented by a judicious choice of time-stepping algorithm.

It should be emphasised that solving the transport equation is usually significantly more computationally expensive than solving the Bateman equations, particularly when using Monte Carlo; even when a problem features many burnable regions, Monte Carlo and depletion tend to differ in expense by at least an order of magnitude [39]. Hence, there is an incentive to use algorithms which require as few transport solutions as possible while maintaining accuracy.

The simplest time-stepping algorithm – although still often applied in reactor physics [71] – should be broadly familiar to any novice of numerical methods: the explicit Euler scheme (or the Predictor scheme, or the Constant Extrapolation scheme (CE)). The scheme is simple: at a point in time with a given nuclide density field, a transport solution is computed. With this, region-wise burn-up matrices are constructed. Using a Bateman solver, each region in the problem is burned to a future point in time, assuming fixed microscopic reaction rates. The process is then repeated for as many time-steps as desired. This logic is shown in Algorithm 1, where N_0 is the initial nuclide density field, n_{steps} is the number of time-steps, Δt is the time-step length (which typically varies across the burn-up period), and n is the time-step index. As an aside, it has been emphasised by Isotalo that, while the scheme is known in the literature as 'the explicit Euler scheme', technically, it is not equivalent to the Euler scheme from calculus in that the nuclide density derivative changes across the time-step (due to the changing nuclide density) [46]. Nevertheless, the name will be used throughout this work.

Input: $N_0, n_{\text{steps}}, \Delta t$
for $n = 0, \ldots, n_{\text{steps}} - 1$ **do**
  $\mathbf{A}_n \leftarrow \psi(N_n)$
  $N_{n+1} \leftarrow \exp\left[\mathbf{A}_n \Delta t_n\right] N_n$
end

Algorithm 1: Explicit Euler method.

The most obvious deficiency with the explicit Euler scheme is that it makes no account of the change in flux spectrum over the course of a time-step. Hence, where the flux spectrum changes significantly, large errors will result, e.g., where highly-absorbing xenon is being rapidly evolved, where burnable absorbers are present, or where a significant portion of fuel is consumed during a time-step. As a result, more sophisticated algorithms are commonly employed, particularly where Monte Carlo is used as the transport solver, placing computational efficiency at a premium.

Chief among these is the Predictor-Corrector scheme (PC), which is also familiar to the wider numerical methods community [79]. There are a number of variants, with one of the most common being the Constant Extrapolation/Linear Interpolation scheme (CE/LI), implemented in Serpent [69], ORIGEN [99], and OpenMC/OpenDeplete [51]. PC methods normally require two transport solutions per time-step – one at the beginning-of-step (BOS) and one at the end-of-step (EOS). In the case of CE/LI, the scheme is initially identical to the explicit Euler scheme (hence the 'CE'), performing a transport solve at the BOS and using it to extrapolate forward in time. At the EOS, a new transport solution is obtained; rather than using it to extrapolate forward in time again, it is instead used to 'correct' the flux calculated at the BOS. The simple CE/LI method burns the BOS nuclide densities forward again but using a BOS-EOS averaged burn-up matrix – the 'Linear Interpolation' portion of the algorithm. The applied burn-up matrix should thus be more representative of the flux across the time-step. The traditional CE/LI scheme is depicted in Algorithm 2. Other depletion solvers, such as BGCORE [35], differ by re-burning from the BOS but using the EOS burn-up matrix, then averaging the two estimates for the EOS nuclide density field. Alternatively, the BOS can be burned forward by $\Delta t/2$, a new transport calculation may be performed, and these reaction rates can be taken as representative across the time-step.

Input: $N_0, n_{\text{steps}}, \Delta t$
for $n = 0, \ldots, n_{\text{steps}} - 1$ **do**
  $\mathbf{A}_n \leftarrow \psi(N_n)$
  $N_{n+1} \leftarrow \exp[\mathbf{A}_n \Delta t_n] N_n$
  $\mathbf{A}_{n+1} \leftarrow \psi(N_{n+1})$
  $\mathbf{A} \leftarrow \frac{1}{2}(\mathbf{A}_n + \mathbf{A}_{n+1})$
  $N_{n+1} \leftarrow \exp[\mathbf{A} \Delta t_n] N_n$
end

Algorithm 2: Predictor-Corrector method.

Although PC schemes double the computational expense per time-step due to the additional transport solution, the increased solution accuracy and feasible increase in time-step length more than justifies the burden due to the method's higher-order [48, 51]. However, burnable absorbers such as gadolinium remain challenging for standard PC methods and have historically required separate treatment [81]. Even without burnable absorbers, the acceptable step-length must often be limited to ensure accuracy – hence, the need for higher-order methods.

Isotalo introduced so-called 'substep' methods to neutronics/depletion coupling [48]. These methods work by dividing the time-step up into multiple substeps, over which nuclide depletion is performed, updating the weighting between the BOS and EOS burn-up matrix for each, in the case of the corrector step. By more accurately representing the change in flux

during a time-step, this has been shown to greatly increase the accuracy of depletion [48, 49]. Sub-stepping can also be applied to the predictor step: by storing a previous BOS burn-up matrix, a linear extrapolation of the burn-up matrix can be made during the predictor, rather than a constant extrapolation. This is designated by Isotalo as an LE predictor, as opposed to a CE predictor. Additionally, instead of performing a linear interpolation during the corrector, the previous BOS burn-up matrix can be used for quadratic interpolation instead, denoted QI rather than LI. While Isotalo recommends using LE over CE wherever possible, the choice between LI and QI is less clear, although the latter is limited by greater memory burden and cannot be used for discontinuous operating conditions between time-steps, e.g., a change in power or a control rod movement [46]. The LE/LI algorithm is shown in Algorithm 3. Here n_{ss} is the number of substeps, s is the substep index, and $w_{n,s}$, $w_{n-1,s}$, and $w_{n+1,ss}$ are the weights for either extrapolation or interpolation applied to the current BOS solution, the previous BOS solution, and the EOS solution, respectively, for the given substep.

Input: $N_0, n_{\text{steps}}, \Delta t, n_{\text{ss}}$
for $n = 0, \ldots, n_{\text{steps}} - 1$ **do**
  $\mathbf{A}_n \leftarrow \psi(N_n)$
  if $n > 0$ **then**
    $N \leftarrow N_n$
    for $s = 0, \ldots, n_{\text{ss}} - 1$ **do**
      $\mathbf{A} \leftarrow w_{n,s}\mathbf{A}_n + w_{n-1,s}\mathbf{A}_{n-1}$
      $N \leftarrow \exp\left[\mathbf{A}\frac{\Delta t_n}{n_{\text{ss}}}\right] N$
    end
    $N_{n+1} \leftarrow N$
  else
    $N_{n+1} \leftarrow \exp\left[\mathbf{A}_n \Delta t_n\right] N_n$
  end
  $\mathbf{A}_{n+1} \leftarrow \psi(N_{n+1})$
  $N \leftarrow N_n$
  for $s = 0, \ldots, n_{\text{ss}} - 1$ **do**
    $\mathbf{A} \leftarrow w_{n,s}\mathbf{A}_n + w_{n+1,s}\mathbf{A}_{n+1}$
    $N \leftarrow \exp\left[\mathbf{A}\frac{\Delta t_n}{n_{\text{ss}}}\right] N$
  end
  $N_{n+1} \leftarrow N$
end

Algorithm 3: LE/LI substep method.

A further extension of substep methods is the log-linear approximation: in strongly self-shielded geometries featuring strong absorbers, the microscopic reaction rates of the absorbing nuclides vary logarithmically with their nuclide density. Accounting for this

variation has been shown to improve the accuracy of problems featuring gadolinium-bearing pins for large time-steps [16, 62].

Given that Monte Carlo calculates statistical estimates of the fluxes and cross-sections which are subject to noise, it has been shown that a bias of a statistical nature will result in burn-up calculations [31]. However, it has been pointed out that this bias is proportional to $\frac{1}{n}$, where n is the number of particles simulated, whereas statistical error is proportional to $\frac{1}{\sqrt{n}}$. Therefore, in most circumstances, this bias should be small in comparison to the simulation noise [46].

This section has dealt with the accurate prediction of nuclide densities with burn-up. However, these details are only generally applicable to small geometries which are neutronically tightly-coupled, i.e., the major dimension of the geometry is not much larger than the neutron mean free path length. Where this is not the case, coupled neutronics/depletion solvers may fail to produce accurate or even physically reasonable answers [26, 28, 50, 51, 55, 56, 60]. Although this problem is most frequently noted in situations where Monte Carlo is the transport solver, it has also been demonstrated for a simple deterministic solver [24] and is suspected as having occurred with a nodal diffusion solver [59]. These are the numerical instabilities with which this thesis is concerned.

1.5 What is 'numerical instability'?

Numerical instability is different from numerical error. Returning an inaccurate solution to a burn-up problem may mean under- or over-predicting the nuclide density of a particularly important isotope and can be reduced by decreasing the time-step length or increasing the order of the time-stepping scheme. Any resulting error is typically localised. Numerical instability, on the other hand, implies a large change in the global solution behaviour which is also driven by the time-step length and scheme. However, its occurrence is typically more binary: for a given scheme, below some threshold time-step length, the solution will take one form – above the threshold it will take another. This is inconvenient as, while attempting to minimise computational cost by maximising time-step length, it is possible that a simulation may prove to be entirely unrepresentative of reality. Monte Carlo users may find this particularly troublesome given that the method is broadly assumed to return an excellent approximation to reality. In computational physics, instability can usually be evaded by decreasing the time-step length or by using an implicit solver. It is perhaps worth emphasising that higher-order solvers do not directly provide stability. One can have a high-order explicit solver which is accurate for large time-steps on stable problems but unstable otherwise, or a low-order implicit solver which is inaccurate for large time-steps on every problem but

provides an assuredly stable solution. In the present context, this will be seen in Chapter 3 and has been reported by Josey and Isotalo et al. [51, 50].

Arguably, the literature on the stability of coupled transport and depletion can be somewhat evasive when it comes to defining exactly what numerical instability is. Two confident descriptions are found in Josey's thesis [51] and in the paper by Densmore et al. [24]. The former describes simple, traditional types of stability which are important for time-stepping schemes (A- and L-stability) as well as the presently more relevant ‘Xenon Instabilities’ where the interactions between xenon and neutronics causes asymmetry to develop in a symmetric problem – the role of xenon will be elaborated on in Section 1.6.1. The latter describes numerical instability in terms of the excitation of higher transport solution eigenmodes. Again this is due to an interaction between neutronics and depletion, although without requiring the presence of a xenon-like isotope – only the presence of a fissile isotope and a relatively weakly-absorbing but stable fission product. Both, however, refer to numerical instability as the result of some perturbation in the problem at hand growing with time to dominate the form of the solution. In the deterministic realm, perturbations often refer to errors induced by a computer's finite-precision arithmetic. Unfortunately, when applying Monte Carlo to obtain a neutronics solution, even larger perturbations are assured due to the presence of stochastic noise.

In many cases, the manifestation of numerical instability need not be intuitively obvious – who is to say what the nuclide distribution should be in a complicated reactor geometry after potentially multiple years of burn-up? Originally numerical instability was noticed as a large oscillation of the power and xenon profiles between time-steps. However, xenon-driven oscillations may occur naturally in real reactors, albeit, normally with a smaller (and hopefully decaying) amplitude, typically assured by the thermal-hydraulic feedback [6]. Therefore, when attempting to interrogate numerical stability, one requires either a reference system with a detailed evolution, or a simple system with known solution properties that may be exploited as a measure of stability.

Physical references are problematic as, even when such a reference may be found, neutronics and depletion are difficult to disentangle from the other physics occurring: inevitably, for a system undergoing any significant burn-up, thermal conditions in the system will also evolve. This precludes validating neutronics and depletion in isolation. Hence, references of any complexity must be numerical.

Numerical references are also challenging, however, as they may be affected not only by the time-step length but also differences in the depletion and transport solvers; there are relatively few sophisticated 3D deterministic solvers which share a depletion engine with a Monte Carlo code. One example is the VERA suite [95], produced by the Consortium

for the Advanced Simulation of Light Water Reactors (CASL), where the 2D/1D transport solver, MPACT [58], shares the ORIGEN depletion engine [21, 99] with the Monte Carlo code Shift [77]. Unfortunately, this suite is under active development and export control – it is not straightforward to obtain. Another example is the ANSWERS suite which contains the WIMS lattice physics code and the MONK Monte Carlo solver [71, 83]. Although readily available, these tools are sadly not viable for thorough examination given that there is little documentation for the depletion solver and the solver is presently limited to the explicit Euler method. Alternatively, one could, in principle, use a Monte Carlo code but take very short time-steps to generate a reference solution. This has been attempted by others: even for simple cases, such as a fuel pin surrounded with a coolant of uniform density, this still results in non-physical oscillations developing [50, 51].

Thus, it seems that the only reliable means of diagnosing the occurrence of instability is to take a relatively simple solution with known properties. The typical cases considered are PWR fuel pins, with a height of 3.66 m and pitch of 1.26 cm, uniform coolant density, imposing reflective boundaries radially and either reflective or vacuum boundaries axially. The former case must have a flat power and nuclide density profile across its length while the neutron flux increases with time, whereas the latter should have power and nuclide density as axially symmetric with time. Admittedly, even if the problem considered is more complex than the fuel pin, as long as it is symmetric then its stability can be definitively examined. Some variant of the uniform or symmetric burn-up problem has been considered by most authors directly addressing the question of burn-up stability [24, 26, 28, 50, 51, 55, 56, 60]. In each of the studies mentioned, numerical instability has been observed to some degree.

Here it is worth showing precisely what instability looks like. This is done by simulating the PWR pin described above subject to a vacuum boundary axially, with a typical power density of 104 W/cm^3, over 180 days of burn-up using a maximum time-step length of 20 days, fuel at 5% enrichment and discretised into 10 axial equal-volume burnable regions. A 20 cm thick homogenised water/steel reflector material surrounds the pin axially, while the fuel has a radius of 0.4095 cm, and the cladding is natural zirconium with a thickness of 0.0655 cm. The coolant density is uniformly 0.7 g/cm^3. The Monte Carlo code Serpent is used to simulate the pin due to its integrated burn-up capabilities [69]. All neutron transport simulations in this work used the ENDF/B-VII nuclear data libraries [17]. The Monte Carlo simulations used 200 inactive cycles, 2000 active cycles, and 30,000 particles per cycle. Simulations are performed using both the Euler and PC schemes and the resulting fission power (in red/yellow) and thermal flux distribution (in blue/white) are shown in Figs. 1.2 and 1.3 with time progressing from left-to-right on each. Serpent's reported maximum statistical

error in the neutron flux for any given region at the initial burn-up step was no greater than 0.23%.

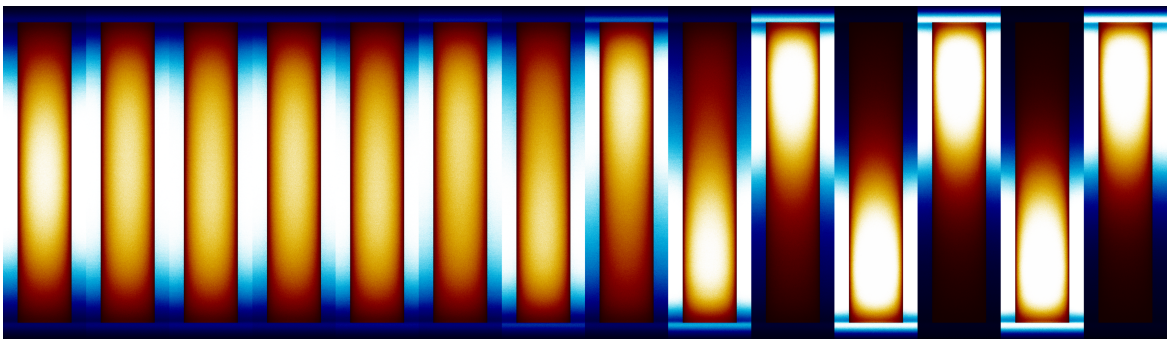


Fig. 1.2 Pin power/flux evolution when using the Euler scheme.

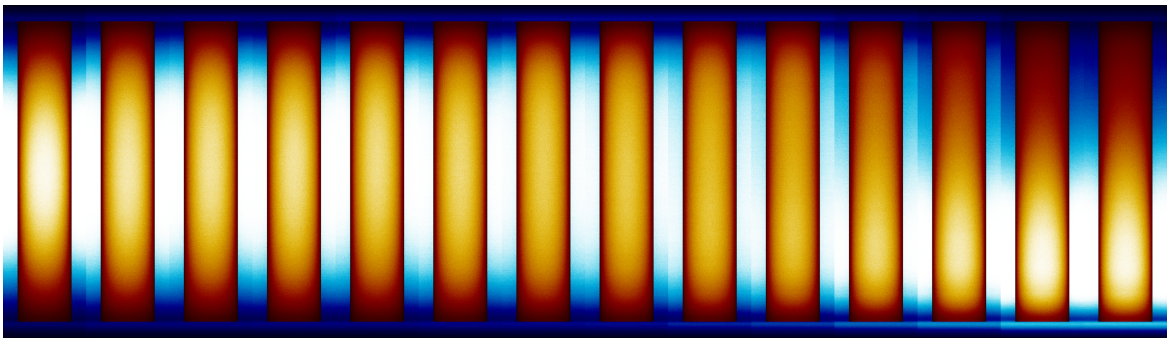


Fig. 1.3 Pin power/flux evolution when using the PC scheme.

The problem is symmetric and should remain so with burn-up (neglecting stochastic noise). Instead, the flux/power distribution in the Euler scheme rather quickly begins to tilt to one side of the geometry and then the other between time-steps. The PC scheme has a less pronounced instability: the swing into asymmetry is more gradual, becoming apparent only in the last few steps. Instability associated with the PC scheme also has the slightly more complex behaviour of not exhibiting power oscillations but rather shifting the power down to one side of the pin over time. The power oscillation is actually still occurring but is obscured behind the corrector step – the swings to opposite sides between predictor and corrector partially cancel each other out during the reaction rate averaged depletion, causing the flux to remain tilted in one direction. In fact, if the corrector step was performed again the oscillatory behaviour would be recovered – this is shown in Chapters 3 and 4. There is nothing privileged about the end of the pin to which the power tilts – different simulations with initial random number seeds will result in the power depositing itself on either side of the pin.

One might reasonably suggest that this appears to be a pathological problem, driven by statistical variations in the Monte Carlo simulation breaking the symmetry. The rebuttal to this claim is that, in the first instance, others have also observed oscillations with similar problems when using many more particles and shorter time-steps, reducing the statistical error while lessening its impact [51]. Secondly, it has been shown that instability can occur even for simple deterministic cases. Namely, Densmore et al. demonstrated the appearance of non-physical solutions modes when using a mono-energetic diffusion solver and two-nuclide depletion chain on a 1D problem with an initially uniform composition and reflective boundaries [24]. Finally, although more difficult to definitively diagnose as instability behaviour, non-symmetric problems have been known to produce questionable results as well: for a PWR assembly with non-uniform coolant density, rather than the power becoming flat over time due to the non-uniform production of fission products, the power may peak or oscillate dramatically, similar to what occurs in Figs. 1.2 and 1.3 [60]. This behaviour can be seen in Chapter 2. This is certainly not to suggest that Monte Carlo noise plays no role in instability phenomenon, but it would be premature to blame it solely.

1.6 Methods for stabilising depletion

High-fidelity simulations are, of course, able to produce reasonable results. Therefore, there must be some numerical means of achieving physically sensible solutions. The two most common techniques are described in this section. Briefly, a third is sometimes suggested: exploiting symmetries within the geometry. This reduces the size of the problem, eliminating higher flux eigenmodes, and shrinking the dominance ratio (described in Chapter 2). However, this will not be applicable to any reactor problem where, say, the coolant density is non-uniform across the core height, or where radial asymmetries exist.

1.6.1 Enforcing xenon equilibrium

Xe-135 has the highest known thermal absorption cross-section of any nuclide, on the order of 2,000,000 b; for comparison, U-235 has a thermal fission cross-section on the order of 600 b. Xe-135 is produced as a minor fission product and as the decay daughter of the major fission product I-135, with a half-life of about 6.5 h. Xe-135 itself has a half-life of 9.2 h. Due to its strong neutron absorption and significant production in fission, xenon density is tightly coupled to neutronics and has a major impact upon reactor kinetics and control [6]. The evolution of xenon and iodine is so fundamental to nuclear engineering that their simplified 0D, mono-energetic kinetics equations are often presented in a first course in

reactor physics. These are:

$$\frac{\mathrm{d}I}{\mathrm{d}t} = \gamma_{\mathrm{I}}\Sigma_{\mathrm{f}}\phi - \lambda_{\mathrm{I}}I \tag{1.22}$$

$$\frac{\mathrm{d}X}{\mathrm{d}t} = \gamma_{\mathrm{X}}\Sigma_{\mathrm{f}}\phi + \lambda_{\mathrm{I}}I - \sigma_{\mathrm{c,X}}\phi X - \lambda_{\mathrm{X}}X \tag{1.23}$$

where X is xenon density, I is iodine density and the other symbols correspond to the typical notation for cross-sections, fluxes, decay constants and fission yields.

Xenon appears to be equally troublesome in burn-up simulations as it has been found that forcing the xenon and iodine concentrations to their time-independent or equilibrium values stabilises coupled neutronics and depletion [50]. This entails zeroing the time derivatives in Eqs. (1.22) and (1.23) such that the equilibrium concentrations may be found as:

$$I = \frac{\langle \gamma_{\mathrm{I}}\Sigma_{\mathrm{f}}, \phi \rangle}{\lambda_{\mathrm{I}}} \tag{1.24}$$

$$X = \frac{\langle (\gamma_{\mathrm{I}} + \gamma_{\mathrm{X}})\Sigma_{\mathrm{f}}, \phi \rangle}{\langle \sigma_{\mathrm{c,X}}, \phi \rangle + \lambda_{\mathrm{X}}} \tag{1.25}$$

Implemented into a Monte Carlo algorithm, the above equations allow the iodine and xenon densities in a given burnable region to be recalculated with each cycle (or over multiple cycles) by using the fluxes and cross-sections accumulated during that period – this has been emphasised by the use of bra-ket notation, showing which integrals the Monte Carlo simulation must compute. The first implementation for enforcing xenon equilibrium in the literature was by Griesheimer [38], although this was done without explicit regard to its application in burn-up problems. Rather, it appears to have been carried out to allow Monte Carlo to efficiently compete with the capabilities of lattice physics codes, many of which have some form of xenon equilibrium treatment.

The stability imparted by enforcing xenon equilibrium is demonstrated in Fig. 1.4, using the same PWR pin as described before. No oscillations are observed over the course of the simulation. It has been noted that burn-up instability has not been reported in fast reactor systems, presumably due to xenon having a much smaller cross-section in the fast spectrum and thus being less tightly-coupled to neutronics [56].

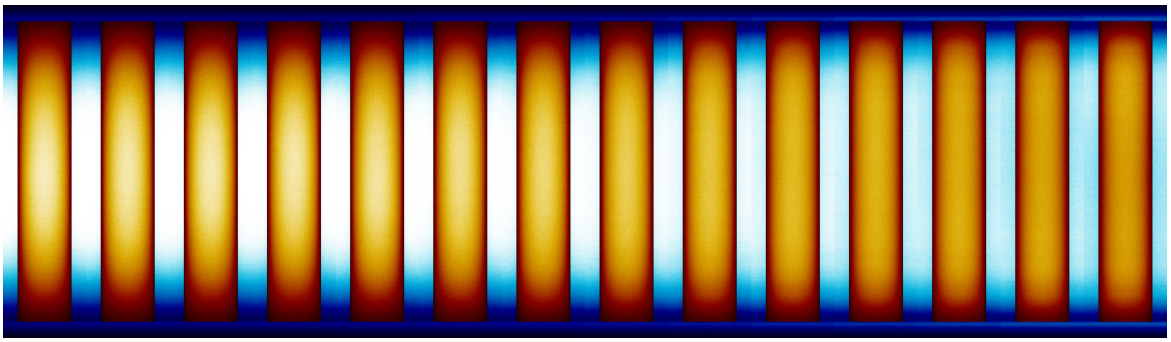


Fig. 1.4 Pin power/flux evolution when using the PC scheme while enforcing xenon equilibrium.

The drawback of enforcing xenon equilibrium is that xenon and iodine's evolution can no longer be followed during a simulation – transients driven by xenon on the scale of hours would be impossible to resolve. Additionally, in the simple problem presented by Densmore et al., there is no xenon analogue, implying that numerical instabilities are possible even if xenon is removed from the calculation [24]. The same is found to be the case by Isotalo et al. [50] who reported oscillations for a pin problem when taking either 30 day or 60 day time-steps – depending on the scheme – even while xenon equilibrium is enforced. This work considers the stability of enforcing xenon equilibrium in Chapter 3.

When presenting the paper on xenon equilibrium at the Serpent User Group Meeting in Berkeley [50], Isotalo also suggested the existence of numerical instabilities which are driven by gadolinium, even while enforcing xenon equilibrium. The mechanism by which this might occur is the following:

1. Assume that two identical regions in a problem contain a strong, fixed, burnable absorber.

2. The flux solutions in each region should be identical but may differ due to, say, the statistical noise or correlation of the Monte Carlo simulation.

3. Hence, one region may have a marginally higher flux and thus burn more during a time-step.

4. On the subsequent time-step, there is less absorber and the true solution in each region is no longer identical – now the correct solution will have a higher flux in the slightly more burned region.

5. Thus, burnable absorber will continue to deplete at a greater rate in this region, establishing a positive feedback and a non-physical flux tilt.

Table 1.1 Isotopic composition of the gadolinium-bearing fuel pin.

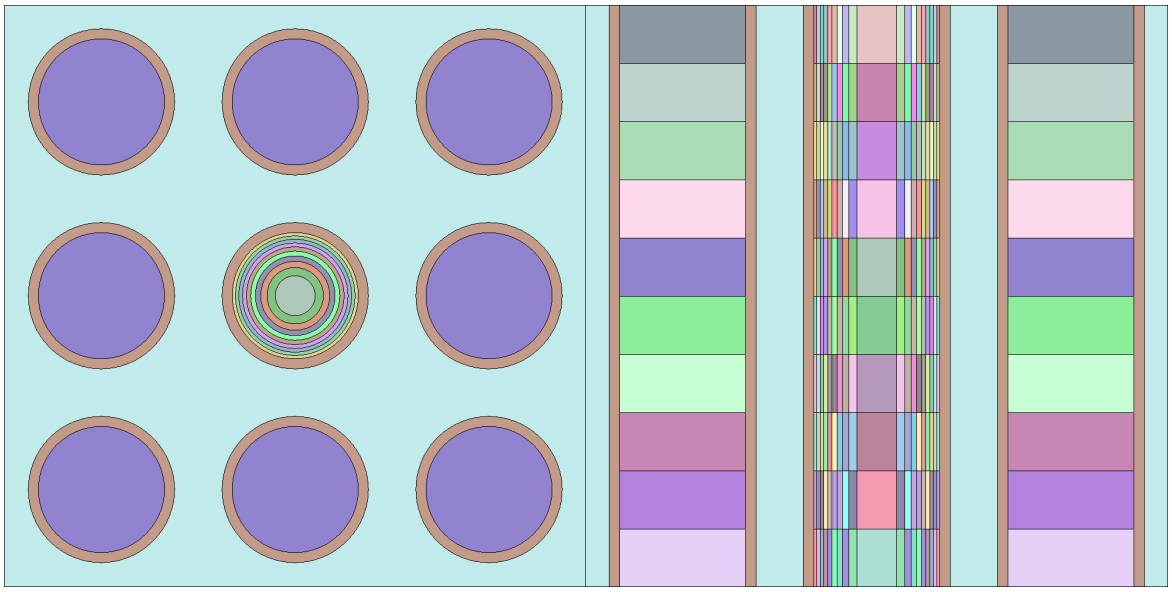
Isotope	**Atomic density (atoms/b/cm)**
U-234	4.2940×10^{-6}
U-235	5.6226×10^{-4}
U-238	2.0549×10^{-2}
Gd-154	4.6173×10^{-5}
Gd-155	2.9711×10^{-4}
Gd-156	4.1355×10^{-4}
Gd-157	3.1518×10^{-4}
Gd-158	4.9786×10^{-4}
Gd-160	4.3764×10^{-4}
O-16	4.5243×10^{-2}

A model was developed in Serpent to investigate the proposed gadolinium instability: the geometry (shown in Fig. 1.5) is a 3×3 lattice of PWR pins. All but the central pin are the 5% enriched UO2 pins described in Section 1.5, while the central pin contains gadolinium – the composition of this pin is taken directly from a Serpent example input and is given in Table 1.1. The pin pitch is 1.26 cm and the height of each pin is 365.76 cm. The coolant density is uniform at 0.7 g/cm^3 and all boundaries are reflective. Each pin is divided axially into 10 burnable regions, while the gadolinium-bearing pin is also divided radially into 10 equal-volume regions. The assembly is burned with a standard PWR power density of 104 W/cm^3 with maximum time-steps of 20 days. The LE/LI scheme is used with 20 substeps and xenon equilibrium is enforced. Each Monte Carlo solution uses 30,000 particles per cycle over 2,000 active cycles and 200 inactive cycles.

Two measures are used to examine the stability of the problem. The first is the eigenvalue: if the same simulation were performed in 2D, i.e., without any axial division, no axial burn-up oscillations can occur and thus the eigenvalue for the 3D case may be compared against this. The 2D simulation was carried out using otherwise identical settings and the resulting eigenvalue was taken as a reference. The difference between the two eigenvalues with time is shown in Fig. 1.6(a). Each simulation had a maximum eigenvalue statistical error of 9 pcm. The second measure assumes that a stable solution will generate a uniform flux axially across a given pin in the mini-assembly. To measure how the flux profile differs from uniformity, the following error metric is used, similar to that found in [26]:

$$\varepsilon = \sum_{i=1}^{n} \left| \frac{\phi_i}{\bar{\phi}} - \frac{1}{n} \right| \tag{1.26}$$

$$\bar{\phi} = \sum_{i=1}^{n} \phi_i \tag{1.27}$$



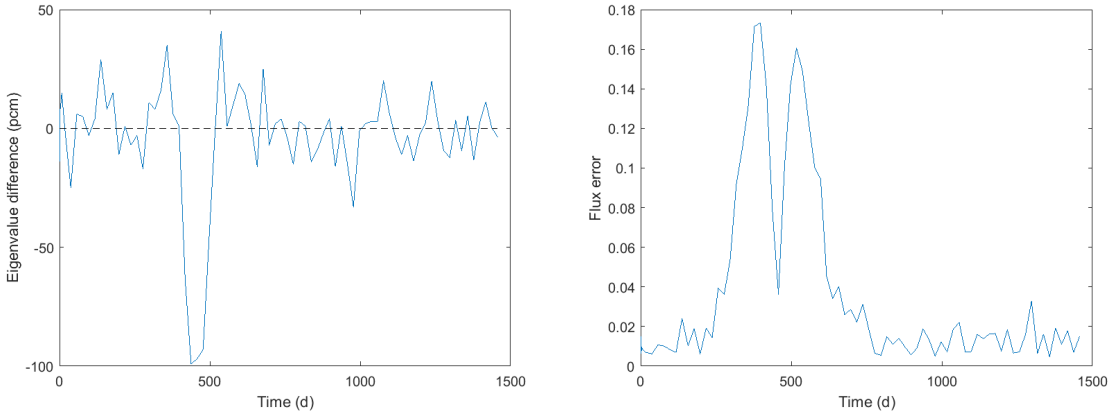
(a) Radial view (b) Axial view

Fig. 1.5 Geometry of the gadolinium-bearing mini-assembly model.

Here n is the number of identical flux tally regions scored and the fluxes, ϕ_i, are integrated across the gadolinium-bearing pins in each axial slice (as opposed to also including the radial flux profiles). These results are shown in Fig. 1.6(b). The maximum statistical error in any estimated flux was 0.26% of the flux's nominal value.

Considering both stability metrics, it is apparent that the 3D simulation becomes significantly erroneous in the vicinity of day 500. The error in the eigenvalue is only relatively slight and brief as compared to the differences in the flux distribution which remains large and substantial for about 500 days of the simulation; previous instability investigations emphasise the difficult of identifying numerical instability by considering the eigenvalue in isolation [28, 60]. This instability is also limited in duration – once much of the gadolinium across the assembly has been consumed the simulation returns to relative uniformity. The strongly axially-peaked power/flux mesh at 400 days is shown in Fig. 1.7 – at this point in time, the ratio of the fluxes in the top and bottom regions of the gadolinium pin is 1.74.

Repeating the same calculations but with axially-vacuum boundary conditions, the non-physical behaviour is no longer observed, with the flux profile remaining symmetric over time. This remains the case when the maximum time-step length is increased to 40 days. Some simulations did begin to show a substantial deviation from symmetry but only due to their fission source being inadequately converged. It may be the case that the reflected system is particularly pathological as opposed to a more realistic configuration.



(a) 2D/3D assembly eigenvalue comparison (b) Non-uniformity of the 3D flux profile

Fig. 1.6 Stability investigation of the gadolinium-bearing mini-assembly with burn-up.

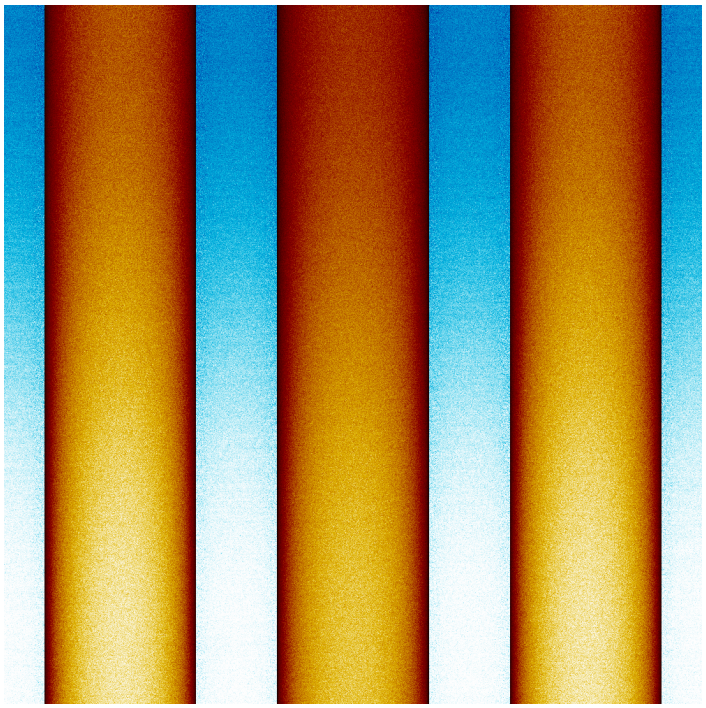


Fig. 1.7 Pin power/flux distribution of the gadolinium-bearing mini-assembly after 400 days of burn-up.

1.6.2 The stochastic implicit Euler method

An alternative approach to spatial stability was developed by Dufek et al.: the stochastic implicit Euler method (SIE) [27]. This is an implicit time-stepping scheme: beginning with a transport calculation and a CE predictor, the corrector transport/depletion step is performed for a specified number of iterations while averaging either the EOS reaction rates or nuclide densities. The original algorithm used only the EOS reaction rates for the corrector step when performing a depletion calculation, causing error accumulation for large time-steps or when using many iterations – this has been remedied by later work on higher-order variants [61, 62, 97]. The mid-point variant (SIMP) is the stochastic implicit equivalent of the PC scheme: its reaction rate averaging algorithm is presented in Algorithm 4 while its nuclide density averaging version is presented in Algorithm 5. An advantage of SI-schemes is that, in principle, they should be agnostic to the nature of the burn-up instability.

Input: $N_0, n_{\text{steps}}, \Delta t, k_{\max}$
for $n = 0, \ldots, n_{\text{steps}} - 1$ **do**
  $\mathbf{A}_n \leftarrow \psi(N_n)$
  $N_{n+1} \leftarrow \exp[\mathbf{A}_n \Delta t_n] N_n$
  for $k = 1, \ldots, k_{\max}$ **do**
    $\bar{\mathbf{A}}^{(k)}_{n+1} \leftarrow \psi(N_{n+1})$
    $\mathbf{A}^{(k)}_{n+1} \leftarrow \frac{1}{k} \sum_{j=1}^{k} \bar{\mathbf{A}}^{(j)}_{n+1}$
    $\mathbf{A} \leftarrow \frac{1}{2}\left(\mathbf{A}_n + \mathbf{A}^{(k)}_{n+1}\right)$
    $N_{n+1} \leftarrow \exp[\mathbf{A} \Delta t_n] N_n$
  end
end

Algorithm 4: Stochastic implicit mid-point method with reaction rate averaging.

Although the algorithms both calculate an average of either the reaction rates or nuclide densities on each iteration, k, it was shown in the original paper that this can also be written as a relaxation. The relaxation factor applied is given by:

$$\alpha_k = \frac{1}{k} \tag{1.28}$$

Relaxation is common in computational physics and especially the numerical solution of partial differential equations: it is used either to 'overcorrect' iterative methods to accelerate convergence or to 'dampen' changes imposed by iterative methods to provide stability. The former technique is referred to as overrelaxation and sets $1 < \alpha < 2$. The stochastic implicit methods and this thesis are concerned with stability and underrelaxation (henceforth referred to as relaxation) where $0 < \alpha < 1$ [79]. The closer the relaxation factor is to 0, the greater

Input: $N_0, n_{\text{steps}}, \Delta t, k_{\max}$
for $n = 0, \ldots, n_{\text{steps}} - 1$ **do**
  $\mathbf{A}_n \leftarrow \psi(N_n)$
  $N_{n+1}^{(0)} \leftarrow \exp[\mathbf{A}_n \Delta t_n] N_n$
  for $k = 1, \ldots, k_{\max}$ **do**
    $\mathbf{A}_{n+1} \leftarrow \psi(N_{n+1}^{(k-1)})$
    $\mathbf{A} \leftarrow \frac{1}{2}(\mathbf{A}_n + \mathbf{A}_{n+1})$
    $\bar{N}_{n+1}^{k} \leftarrow \exp[\mathbf{A} \Delta t_n] N_n$
    $N_{n+1}^{(k)} \leftarrow \frac{1}{k} \sum_{j=1}^{k} \bar{N}_{n+1}^{(j)}$
  end
  $N_{n+1} \leftarrow N_{n+1}^{(k_{\max})}$
end

Algorithm 5: Stochastic implicit mid-point method with nuclide density averaging.

the stability imposed, albeit at the cost of slower convergence. On the other hand, a lighter relaxation where α is close to one may not produce a stable solution.

In the original paper this averaging is described as an advantage because it allows all solutions to contribute equally to the final result. However, this is not an inherently worthwhile outcome. In fact, when performing a relaxation, one typically seeks to maximise the relaxation factor to allow for faster convergence – the smaller its value, the more prolonged is the convergence. Relaxation is covered in more detail in Section 3.2.

This variable relaxation factor has been applied to coupling Monte Carlo neutronics with thermal-hydraulics as well [94]. One apparent benefit of this is, when an error criterion between iterations is employed, the SIE may appear to have a much better error convergence as compared to using a fixed relaxation factor. This is actually somewhat deceptive: all Monte Carlo solutions will be subject to a basic level of noise, limiting convergence. However, if the relaxed variable (either the temperature or nuclide density field) is obtained by a deterministic solver, SI schemes will ensure that they converge to some fixed vector. This is because the relaxation factor approaches zero after many iterations, decreasing the marginal contribution to the final value of the relaxed field over time. Hence, one should naturally expect excellent 'convergence' behaviour compared to other schemes which might be subject to even a small degree of noise.

As mentioned, SI schemes perform a specified number of iterations on the EOS. There is currently no widely adopted convergence criterion or heuristic for deciding how many iterations should be performed although cases in the literature – as well as the users of Serpent – seem to adopt 10 iterations at the very minimum to achieve stability on the likes of a fuel pin or assembly problem [27, 51, 62, 63]. One might imagine many iterations on a Monte

Carlo solution for a single time-point to pose a significant computational challenge, but this is seemingly not as dramatic as might be expected: Serpent provides the option of using fewer particles and cycles during the EOS iteration and even using the source distribution from the previous iteration to reduce the number of inactive cycles.

The SIE's performance is demonstrated in Figs. 1.8 and 1.9 when using 5 and 10 iterations, respectively, for the same pin problem as above. The number of active histories simulated during each corrector iteration was reduced by a factor of 5 or 10 by reducing the number of generations. The number of inactive histories remained the same on the first corrector iteration but were subsequently reduced by a factor of 4, justified by passing the fission source distribution from one iteration to the next to begin from a nearly converged solution.

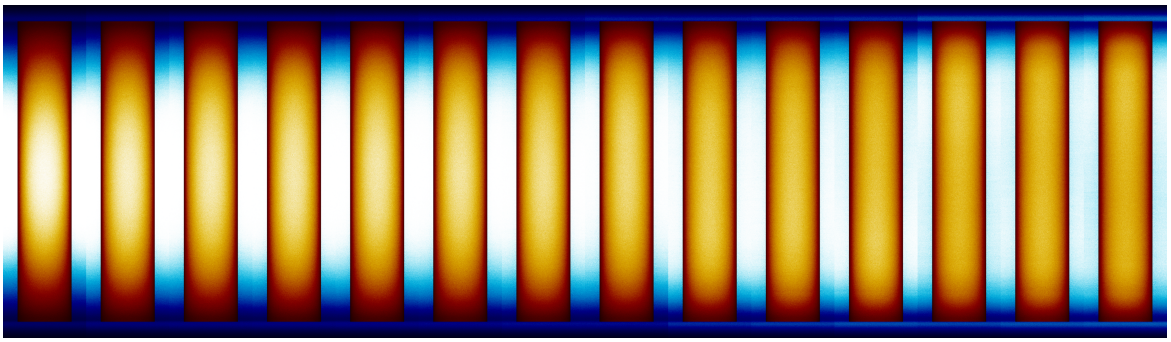


Fig. 1.8 Pin power/flux evolution when using the SIE scheme with 5 iterations.

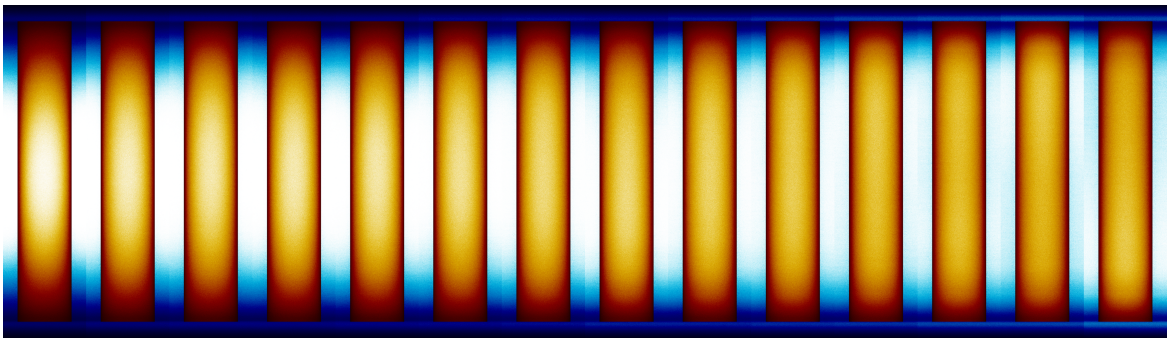


Fig. 1.9 Pin power/flux evolution when using the SIE scheme with 10 iterations.

Although both solutions appear relatively stable as compared to Figs. 1.2 and 1.3, there is, in fact, a mild flux asymmetry beginning to develop just noticeably by the final time-point in Fig. 1.9. This is intriguing as it appears to suggest that more SIE iterations can worsen the solution, at least when attempting to preserve the total active histories simulated at the EOS. To highlight this, the relative errors of the fluxes in axially symmetric nodes are shown in Fig. 1.10, and compared against the simulation above where xenon equilibrium was enforced.

The numbering of the symmetric nodes runs from the outermost (1) to innermost (5). At all but the centre-two nodes, there is a greater relative error when using the SIE (with 5 or 10 iterations) than when simply enforcing xenon equilibrium. The greater peripheral error may be partially attributable to the vacuum boundaries terminating neutron histories, resulting in fewer events being scored, increasing statistical error [92]. However, instability is typically observed as the growth of an anti-symmetric mode, also contributing to the greater error towards the periphery – seemingly quite strongly when using the SIE with 10 iterations. The only obvious suggestion for why the SIE with 5 iterations performs better than that with 10 is due to scoring higher statistics during each corrector transport solution, to which the stabilising properties may be sensitive. Perhaps more iterations would stabilise the problem successfully provided greater statistics were used for each corrector step. Nevertheless, it appears that the optimal number of iterations and necessary computational effort is difficult to determine a priori.

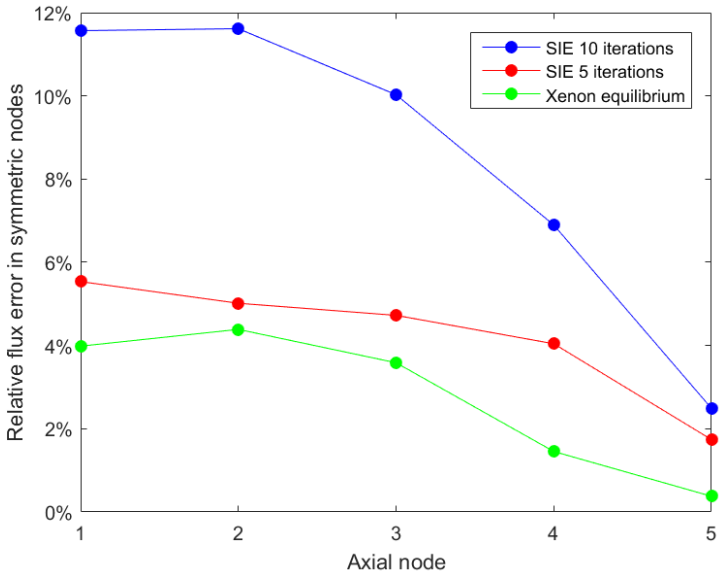


Fig. 1.10 Relative error in symmetric flux values for a PWR pin burned for 180 days when using the SIE scheme or enforcing xenon equilibrium.

1.7 Statement of objectives and thesis organisation

Both the stochastic approximation and enforcing xenon equilibrium work to stabilise depletion problems, although neither is wholly reliable. This stems, in part, from the nature of burn-up instabilities not being entirely understood – most previous studies merely highlight the existence of these instabilities under various conditions, often attempting to isolate their

cause only by varying time-step lengths or applying a different depletion scheme. The basic objective of this thesis is, therefore, to better understand what causes this non-physical behaviour. In doing so, a second objective may be achieved: it may be possible to suggest more efficient means of dealing with challenging burn-up problems and to articulate when or why different methods might fail. The resulting thesis is predominantly targeted towards the developers of Monte Carlo reactor physics software and those intimately familiar with its use in complex problems. That being said, it is hoped that the foregoing discussion on the occurrence and avoidance of instability – as well as subsequent emphasis on judicious choice of Monte Carlo settings – will be of some use to the average Monte Carlo code user.

The first attempt to interrogate the nature of burn-up instabilities is made by adding additional parameters to earlier burn-up studies, namely, the Monte Carlo settings. Previous works rarely – and, sometimes, dismissively – address the role of the Monte Carlo solver in causing burn-up instabilities, although none have explicitly investigated this. Chapter 2 aims to fill this gap by exploring how Monte Carlo settings interact with time-step, coupling scheme, and geometry in causing instabilities. This will attempt to determine what – if any – role they play in previous reports of non-physical behaviour. This chapter largely follows the author's first publication [20].

With insights gained from Chapter 2, Chapter 3 attempts to address the second objective by optimising the use of implicit schemes for achieving stable solutions with minimal computational expense. The stochastic implicit methods are the de facto stable coupling schemes applied to Monte Carlo neutronics and depletion, but they are neither unique nor necessarily optimal. Hence, variations on implicit methods are explored and their performances are discussed. This chapter follows the author's second publication [19]. The effect of enforcing xenon equilibrium alongside an iterated corrector step is also investigated to examine how different 'stable' depletion methods interact.

To-date, perhaps the most thorough theoretical analysis of depletion instabilities has been performed by Densmore et al. [24], and Chapter 4 follows in that vein. By extending their analysis, even for a simple system, additional understanding of burn-up instabilities and how they manifest with different coupling schemes can be attained.

Chapter 5 summarises the thesis findings, proposes an explanation for the various occurrences of burn-up instabilities, and suggest future work in coupling Monte Carlo with depletion and additional physics.

Chapter 2

Neutron clustering as a driver of burn-up instability

The problems in which burn-up instabilities are noted to occur are typically large, spatially-decoupled, and possess a high dominance ratio. These properties have long been known to pose a challenge for Monte Carlo methods. This chapter discusses the recently discovered phenomenon of neutron clustering – noted to occur in such problems – and examines how it affects the behaviour of burn-up calculations on long and short time-scales.

2.1 Correlation in Monte Carlo neutron transport

Monte Carlo methods have long been known to under-estimate the error of their results [9]. This is due to the phenomenon of 'inter-generational correlation'. When accumulating statistics for the eigenvalue, reaction rates, etc., most Monte Carlo codes assume that all samples of an event occur independently of one another. This is not strictly true due to the generational algorithm used wherein one generation of particles produces the subsequent generation, i.e., particle generations are correlated. This means that the central limit theorem no longer applies and the true standard deviation does not decrease as $\frac{1}{\sqrt{n}}$, where n is the number of generations. The extent of this correlation is determined by a given problem's dominance ratio. The dominance ratio is defined as the ratio of the fundamental and first eigenvalues of the problem:

$$\rho = \frac{k_1}{k_0} \tag{2.1}$$

where k_0 is the fundamental eigenvalue, k_{eff}. The closer the dominance ratio is to unity, the greater the correlation between generations. For one-dimensional, one-group diffusion problems with reflective boundaries, it is quite straightforward to estimate the dominance

ratio – more complex problems require numerical methods. For PWR pins, the dominance ratio can be on the order of 0.95 or greater [30], whereas, for a PWR core (like the BEAVRS benchmark [44]), the dominance ratio is on the order of 0.996 [42]. The non-ideal convergence of Monte Carlo error estimates is thoroughly analysed and discussed in Herman's PhD thesis [42].

A recently revealed aspect of correlation in Monte Carlo simulation is the presence of spatial correlations, or neutron clustering, as well as the above discussed 'temporal correlations' [22, 29, 32, 76, 93, 102]. Put simply, during a Monte Carlo simulation, neutrons may die anywhere, either through capture or leakage. On the other hand, where neutrons are born, they are born together in fission, which produces two or more neutrons at each occurrence. In the first generation of the simulation there is no spatial correlation between neutrons, each having been placed by sampling from an initial guess distribution. Spatial correlation is introduced as multiple neutrons are born together in subsequent generations. In the extreme case, a Monte Carlo simulation may begin with the correct fission source distribution but ultimately converge towards an entirely different distribution as the spatial correlation begins to dominate over many generations [32, 93].

It has been demonstrated by both Dumonteil et al. and Sutton and Mittal that simulating more particles per generation tends to reduce the degree of spatial correlation. This was derived for a continuous time representation of the Monte Carlo algorithm, as well as for a more exactly analogous generation-based problem [32, 93]. How many particles one should use to reduce the effects of clustering is dependent upon the system being simulated. Intuitively, the occurrence of clustering can be understood as depending upon the ease with which a neutron may traverse the system and the density of neutrons per unit volume. If a neutron family can traverse a geometry in few generations then relatively few neutrons will be required to prevent under-sampling of the problem. This depends on both the geometric size of the system and its material properties, or the macroscopic cross-sections of the materials.

Encoding the above logic, Dumonteil et al. use a 'hand-waving' (in their own words) means of estimating the number of neutrons one should simulate which is reported as being in accordance with their Monte Carlo simulations [32]. The prescription – for a problem with characteristic length L and neutron mean free path l – is that one should simulate many more than N particles where N is given by:

$$N = (L/l)^3 \tag{2.2}$$

Therefore, when simulating a 4 m tall PWR pin where the mean free path is approximately 6 cm, many more than 300,000 particles per generation are necessary to avert clustering. With

this number in mind, it is elucidating to examine previous reports of numerical instability in coupled Monte Carlo neutronics/depletion simulations.

That said, DeMulatier et al. [22] derive a different estimate for geometries with reflecting boundaries, subject to neutron diffusion, and accounting for the effect of population control in the Monte Carlo algorithm:

$$N = (L/M)^2 \tag{2.3}$$

In this equation, M^2 represents the neutron migration area, which, for a a PWR, is about 56 cm^2. Using these values, for the same 4 m pin, one should simulate on the order of 10,000 particles per generation. On the other hand, Nowak et al. [76] find about 500,000 particles per generation adequate to quench clustering in such a problem. Even so, this estimate is based on attaining an acceptably small standard deviation of the neutron centre of mass during the Monte Carlo simulation – it is not clear how to determine the extent of fluctuation that is acceptable for a given problem.

Another aspect of clustering directly relating to Monte Carlo settings was provided by Sutton & Mittal. In particular, simulating additional cycles tend to worsen the results obtained due to increasing correlation [93]. This increased correlation stems from more neutron 'families' dying off over generations such that, by the conclusion of the simulation, it is possible that all neutrons descend from only one neutron in the initial guess population. Although such an occurrence is extreme, excessively many generations may negatively affect simulation results.

2.2 A survey of prior depletion studies

A number of authors have reported Monte Carlo burn-up instabilities; Table 2.1 summarises some prominent studies focused specifically or substantially on instability. A few additional points of clarification are needed. First, all geometries considered are either PWR pins or assemblies with standard power densities of 104 W/cm^3 and radially reflective boundaries – axially, the boundaries were either vacuum and reflective.

Second, all problems used between 8 and 16 axially-discretised regions. Differences in discretisation will certainly affect the accuracy of the calculations but, presently, there has not been much suggestion that the degree of discretisation affects the stability of the problem – excluding the extreme case of a single burnable region.

Third, all but two problems considered used a uniform coolant density distribution, for ease of stability diagnosis. The two which used a realistic, non-symmetric coolant profile were Dufek et al. (2013) and Kotlyar & Shwageraus (2013) [28, 60]. The former study considers two geometries: it is only the assembly which has a non-symmetric coolant density.

Table 2.1 Detailed descriptions of depletion studies.

Study	Particles per cycle	Active cycles	Inactive cycles	Geometry	Scheme	Unstable step (d)
Dufek & Hoogenboom (2009) [26]	1,000	5,000	1,000	4 m pin, reflective	Euler	2
Dufek et al. (2013) [28]	5,000 & 300,000	30,000 & 400	1,000 & 500	3 m pin, reflective & 3.66 m assembly, vacuum	PC	14 & 5
Kotlyar & Shwageraus (2013) [60]	300,000	1,200	Not specified	3.66 m assembly, vacuum	Euler & PC	5 & 50
Isotalo et al. (2013) [50]	5,000	5,000	1,000	4 m pin, reflective	PC	1
Kępisty & Cetnar (2015) [56]	5,000	1,600	400	3.66 m assembly, vacuum	Euler	5
Josey (2017) [51]	32,000	2,500	2,500	4 m pin, reflective & vacuum	Euler & PC	7

As discussed in Section 1.5, this makes the diagnosis of instability difficult and is presumably why the two studies examining the same problem and applying very similar statistics differ on the value of the unstable time-step. Examining the papers, the former notes an oscillation in the flux and xenon density when 5 day time-steps are used, although this is ambiguous in that it may be a physical phenomenon. More convincingly, the latter identifies a divergence between the eigenvalues obtained when increasing the time-step length from 25 days to 50 days.

With these caveats covered, and noting that, by the estimate of Dumonteil et al., one requires many more than 260,000 particles per generation for a 3.66 m PWR pin, it is clear that each of the previous works have used an insufficient number of particles to eliminate the effects of clustering. Indeed, a PWR pin with reflective axial boundaries is precisely the geometry featured in the Dumonteil et al. paper first highlighting the presence of clustering in Monte Carlo simulations [32]. The only exceptions may be Dufek et al. and Kotlyar & Shwageraus [28, 60] who have used nearly as many particles as recommended by Nowak et al. (500,000) [76], albeit for problems with vacuum boundaries, for which this number may be an underestimate. There is also relatively little consistency between the reported unstable time-step lengths, implying a stochastic component to the instability. That being said, some studies only used a single step-length without varying the value to observe a change in behaviour.

2.3 Investigating the effects of clustering in typical burn-up problems

To investigate the effects of clustering on burn-up, a number of coupled simulations were performed where, for each Monte Carlo solution, the same total number of particle histories were followed while the number of particles per cycle was varied. This was done while varying the total particles histories and only the active particle histories; as Sutton and Mittal have shown, simulating excessively many generations tends to increase the degree of correlation [93]. Therefore, preserving the total number of histories while varying the population size would result in simulating many inactive cycles. Experience suggests that the number of inactive cycles required for the Shannon entropy to reach stationarity is independent of the number of particles simulated per generation. Hence, simulating further inactive cycles only serves to penalise simulations using fewer particles per generation. If neutron clustering does not drive non-physical results, then both sets of simulations should be similarly affected due to simulating the same number of active particles and, thus, achieving similar results with the same degree of statistical error. On the other hand, if clustering does play a role, then the set of simulations following the same total number of histories should attain a greater asymmetry by inducing clustering to a greater extent.

The geometry considered was a PWR pin, identical to that in Section 1.5. However, two versions were simulated: the first had a uniform coolant density, as described previously – the other used a non-uniform coolant density, adapted from [60] and shown in Fig. 2.1. The different coolant density profiles were used in order to determine whether clustering is important only as a mechanism for breaking symmetry. If so, varying the number of particles per generation would produce different results only for the uniform coolant density problem.

The number of particles per cycle were 3,000,000, 300,000, and 30,000. Correspondingly, when preserving the total histories simulated, the number of active/inactive cycles used were 50/100, 500/1,000, and 5,000/10,000. When preserving only the number of active histories simulated, the number of inactive cycles for all simulations was instead set to 100.

Both the Euler and PC schemes were applied to two burn-up schedules – one with a maximum time-step of 20 days, the other with a maximum time-step of 40 days. Each burn-up simulation was performed three times with a different seed to account for differences due to stochastic phenomena. This is, admittedly, relatively few simulations – this was due to constraints on computational resources. Nevertheless, the number of simulations appears sufficient to highlight some qualitative trends.

All simulations were performed using Serpent v2.1.30. To accelerate burn-up calculations, Serpent allows passing the fission source from one time-point to another, in order to reduce

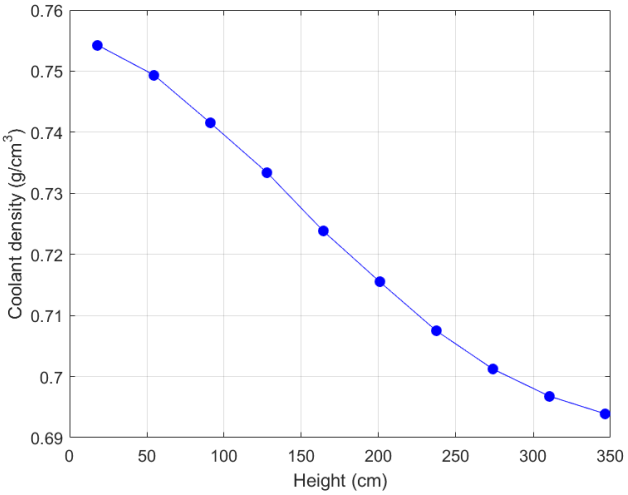


Fig. 2.1 Realistic coolant density profile for a PWR pin.

the number of inactive cycles necessary to achieve source convergence. This was not done here in case the fission source from one time-point should be contaminated by clustering and correlation – all simulations, at all time-points, at the BOS and EOS, began from a uniform source distribution guess.

Both problems require some metric allowing assertions to be made about their stability under different circumstances. For the uniform coolant density pin, the asymmetry of the problem was used; as mentioned previously, the problem should remain symmetric if stable. Therefore, solutions which begin to show some degree of asymmetry must be erroneous. This can be measured with any quantity produced by the simulation, such as a nuclide density. Here it is measured using the one-group scalar fluxes in each region. Specifically, at a given point in time, the asymmetry, ε, is given by:

$$\varepsilon = \sum_{i=1}^{5} \frac{|\phi_i - \phi_{11-i}|}{\max(\phi)} \tag{2.4}$$

where ϕ_i is the scalar flux in region i, $\max(\phi)$ is the maximum value of the flux at the given time-point, and $|\cdot|$ denotes the magnitude. For a non-symmetric problem, there is no property of the solution which may be exploited as an error measure; if two solutions differ, one cannot definitively say which is correct. In this chapter, one solution will be chosen as a 'reference' and the behaviour of other solutions will be compared against it using the normalised L_1 error:

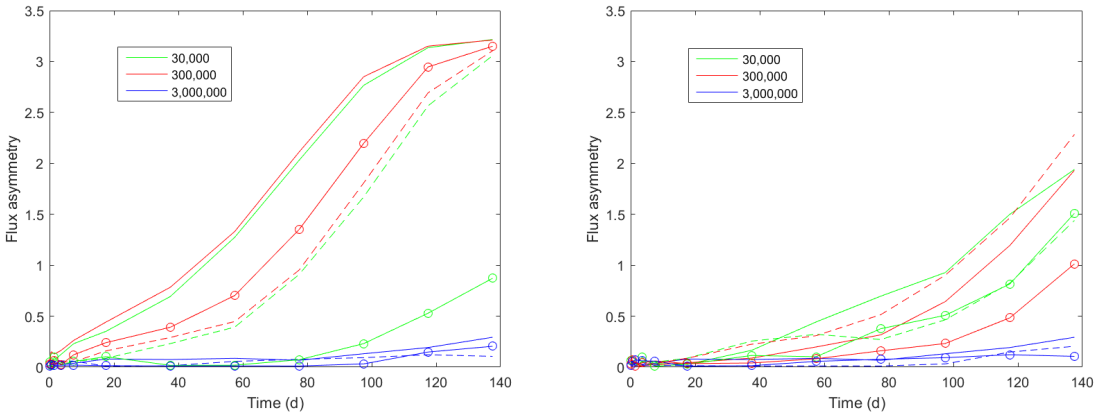
$$\varepsilon = \sum_{i=1}^{10} \frac{|\phi_i - \phi_i^{\text{ref}}|}{\max\left(\phi^{\text{ref}}\right)} \tag{2.5}$$

where ϕ^{ref} denotes the reference solution. The reference used will be one of the 3,000,000 particle simulations given that they should be least contaminated by clustering.

2.3.1 Uniform coolant density PWR pin

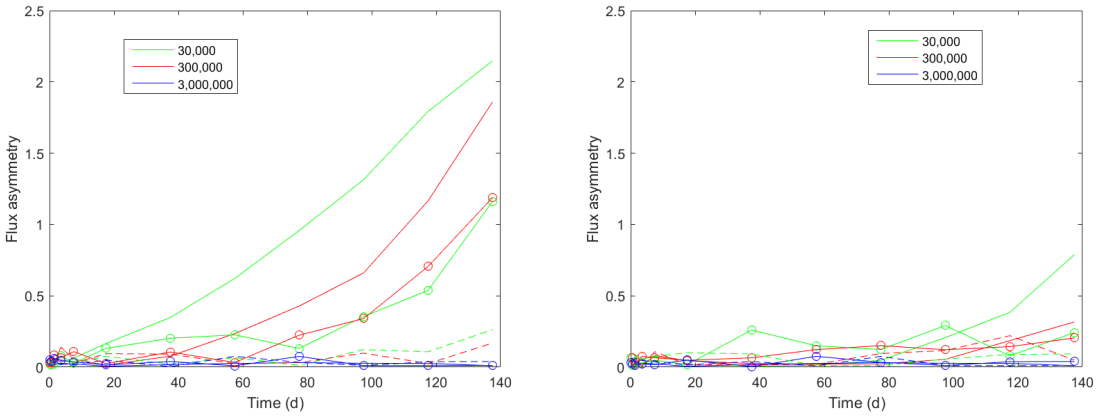
The flux asymmetry for the explicit Euler and PC simulations using 20 day maximum time-steps are shown in Figs. 2.2 and 2.3, respectively. Note that each line colour corresponds to a different number of particles per generation, as shown in the legend, while each line style (unbroken, circled, or dashed) represents a different initial seed used for the given number of particles per generation. This convention is used throughout the remainder of Section 2.3.1. and through Section 2.3.2.

The first observation is that, whether for Euler or PC, using 30,000 or 300,000 particles per generation produces quite variable results. Nearly all simulations, with both schemes, reach a significant state of asymmetry, although the ascent in asymmetry begins at different times, with variable rates. For example, the green circled simulation remains quite symmetric for a long period before becoming only slightly asymmetric towards the end. Even so, there is still striking similarity between otherwise completely different runs: note, for example, the red and green dashed lines in Fig. 2.2(a) which track each other remarkably, despite using different numbers of particles per generation. Regardless of when the asymmetry growth begins, it appears that all asymmetric solutions converge to the same state – inspecting the final flux profiles seems to confirm this, the main differences between the solutions being whether or not they are in phase with each other. This is apparently random and independent of the number of particles per generation: the same behaviour was observed qualitatively when running simulations with fewer particles and lower statistics, with neither side of the geometry obviously privileged. This is demonstrated in Fig. 2.4, showing the final flux profiles in Fig. 2.2(a). That said, there is a difference between PC and Euler in that the former appears more stable over a number of runs given the extent of asymmetry is not as large as with Euler.



(a) Total histories (b) Active histories

Fig. 2.2 Flux asymmetry with time when depleting a PWR pin with uniform coolant density using the explicit Euler scheme, with a 20 day maximum time-step, and varying the number of particles per generation while preserving the total particle histories (left) and active particle histories (right).



(a) Total histories (b) Active histories

Fig. 2.3 Flux asymmetry with time when depleting a PWR pin with uniform coolant density using the PC scheme, with a 20 day maximum time-step, and varying the number of particles per generation while preserving the total particle histories (left) and active particle histories (right).

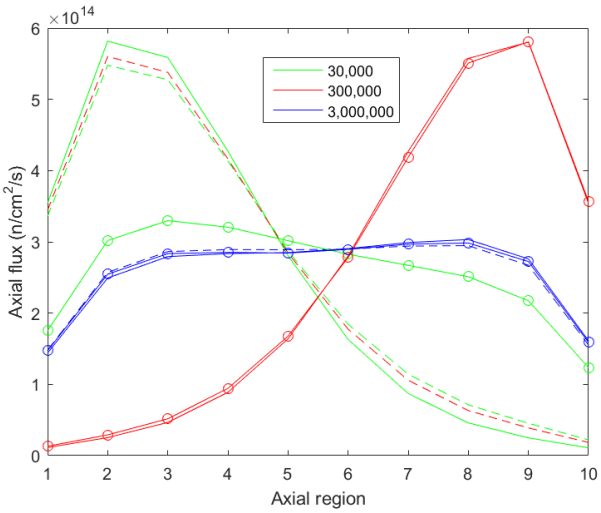


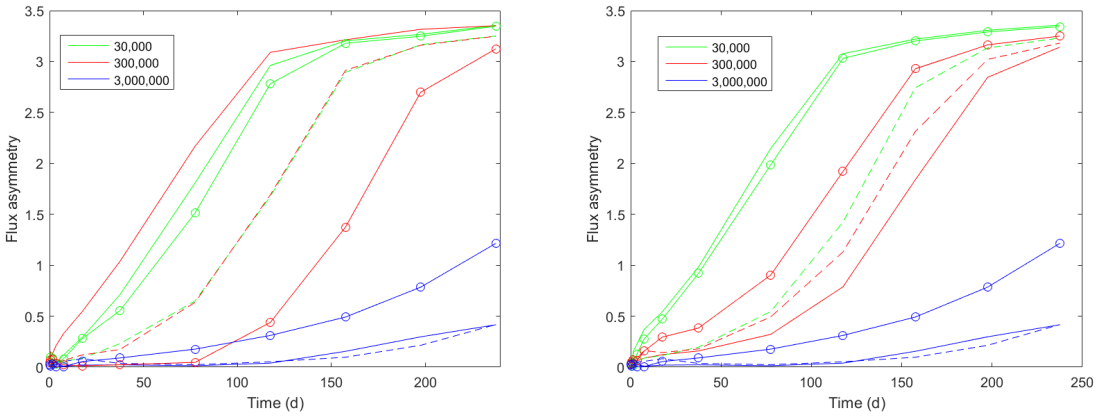
Fig. 2.4 Final axial flux profiles corresponding to Fig. 2.2(a).

It is also noticeable from Figs. 2.2(b) and 2.3(b) that the asymmetry growth is much reduced by preserving only active histories, simulating the same number of inactive cycles across all runs, whether for the Euler or the PC scheme. There are two explanations for this: the first is that, although 100 inactive cycles are adequate for converging the source distribution when the problem is symmetric (and relatively flat due to fission product poisoning), when asymmetric solution modes are excited, this may no longer be sufficient. Therefore, the results may be partially contaminated by the flat source guess distribution, dampening the asymmetry growth. On the other hand, this may be partially attributable to the predictions of Sutton and Mittal wherein simulating excessively many particle generations will tend to exacerbate clustering [93]. In future, this should be addressed using a source convergence acceleration method, minimising the number of inactive cycles necessary while also achieving adequate convergence from an initial uniform source guess.

There is also the observation that using many particles per generation does appear to result in relatively stable, symmetric solutions. However, there is an interesting occurrence for the 3,000,000 particle simulations in Fig. 2.2(a): over time, the asymmetry grows, albeit much more slowly than any of the cases using fewer particles per generation. The most obvious explanation is that the explicit Euler scheme applied to this problem and using a time-step length of 20 days is numerically unstable – as in the simple case described by Densmore et al., higher solution modes are gradually excited over time [24]. This is as opposed to the cases using fewer particles which produce significantly biased transport solutions, falling into the asymmetric state much more quickly. On the other hand, numerical instability does not appear to be a problem for the PC scheme for this problem with a time-step length of 20

days. However, it may simply be that the growth of other solutions modes is so gradual in this case as to be unnoticeable in the number of time-steps taken.

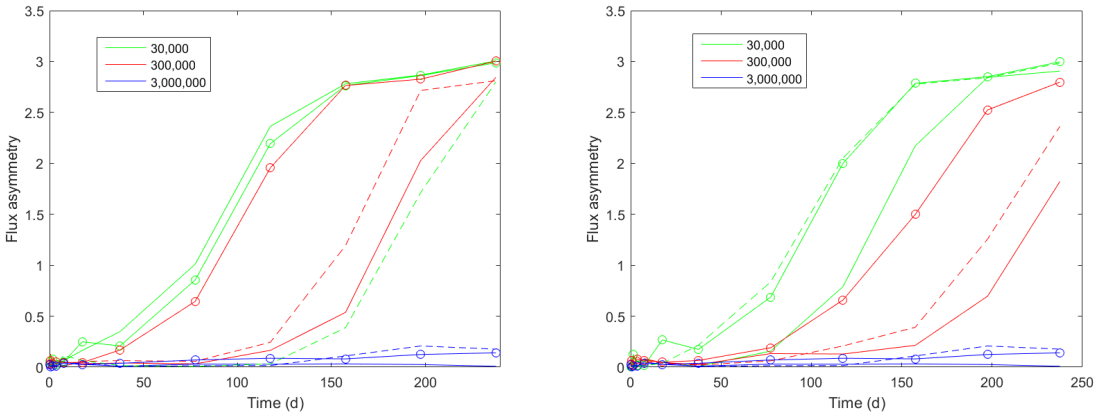
The flux asymmetries for the explicit Euler and PC schemes, when the maximum time-step length is 40 days, are shown in Figs. 2.5 and 2.6, respectively. With either scheme, the behaviour appears broadly consistent with the 20 day time-step cases. The most interesting discrepancy may be the behaviour of the 3,000,000 particle cases with the Euler scheme, each showing substantial asymmetry growth with time. The circled case even shows greater asymmetry than a 300,000 particles run for about the first 80 days of simulation, although the latter suddenly becomes asymmetric thereafter. This lends credence to the suggestion above that two effects drive Monte Carlo depletion simulations towards non-physical solutions: one stochastic and often driven by clustering, the other essentially deterministic, causing a gradual magnification of error, even at the level of floating point arithmetic.



(a) Total histories (b) Active histories

Fig. 2.5 Flux asymmetry with time when depleting a PWR pin with uniform coolant density using the explicit Euler scheme, with a 40 day maximum time-step, and varying the number of particles per generation while preserving the total particle histories (left) and active particle histories (right).

Another difference between the 20 day and 40 day schemes is apparent from the PC results. In all cases where 20 day steps were taken, by the end of the simulation, the asymmetry in cases using fewer than 3,000,000 particles per generation was still growing. With 40 day steps on the other hand, the asymmetry develops more readily, apparently plateauing in some cases, despite there only being one additional burn-up step. Therefore, even if numerical stability precisely in the vein of Densmore et al. [24] is not a concern (given the 3,000,000 particle simulations appear symmetric), larger time-steps more readily excite a clustered simulation into producing non-physical solutions.



(a) Total histories

(b) Active histories

Fig. 2.6 Flux asymmetry with time when depleting a PWR pin with uniform coolant density using the PC scheme, with a 40 day maximum time-step, and varying the number of particles per generation while preserving the total particle histories (left) and active particle histories (right).

Comparing the preservation of total and active histories when taking 40 day time-steps, there is an interesting occurrence: when preserving total histories, there is little difference between simulations using 30,000 or 300,000 particles per generation with respect to the onset and extent of asymmetry. However, preserving active histories, Figs. 2.5(b) and 2.6(b) seem to show that the 300,000 particles per generation simulations are not as prone to asymmetry as the 30,000 simulations. In particular, the 300,000 particle simulations broadly tend to become asymmetric at a later time than the 30,000 particle simulations. This may be due to more particles and fewer generations sufficiently lessening the extent of correlation in the 300,000 particle simulations when preserving active histories – when preserving the total histories, the factor of 10 difference in the number of inactive cycles will have increased correlation, perhaps to the extent to negate the larger number of particles. However, this would require many more simulations to assert definitively.

2.3.2 Non-uniform coolant density PWR pin

In these investigations, one modification was made to the prescribed particle settings given above: as the Shannon entropy for the 3,000,000 particle cases did not appear fully converged within 100 inactive cycles, the number was increased to 200. This is not imagined to significantly affect the conclusions given that subsequent time points (where fission products have flattened the source profile) did converge within 100 inactive cycles and the final

results appeared qualitatively identical. The same was done for the other simulations which preserved only the number of active histories.

For 20 day maximum time-step length, both the flux errors as defined by Eq. (2.5) and the end-of-burn-up flux profiles are presented for the explicit Euler scheme in Figs. 2.7 and 2.8. For the PC scheme, these are presented in Figs. 2.9 and 2.10. The end-of-burn-up flux profiles are presented in order to highlight differences in the flux profile between different cases, given that properties of the true flux profile cannot be deduced a priori. Note that the blue dashed line does not appear in graphs featuring the error, but does appear in those for the flux profile. This is because it is the solution chosen as the reference profile when applying Eq. (2.5).

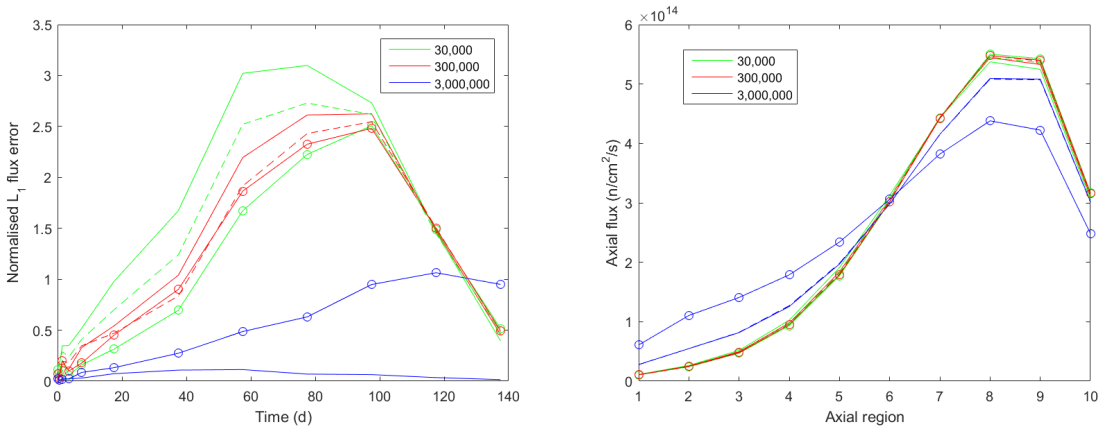


Fig. 2.7 Relative flux errors with time (left) and final flux profiles (right) when depleting a PWR pin with non-uniform coolant density using the explicit Euler scheme, with a 20 day maximum time-step, and varying the number of particles per generation while preserving the total particle histories.

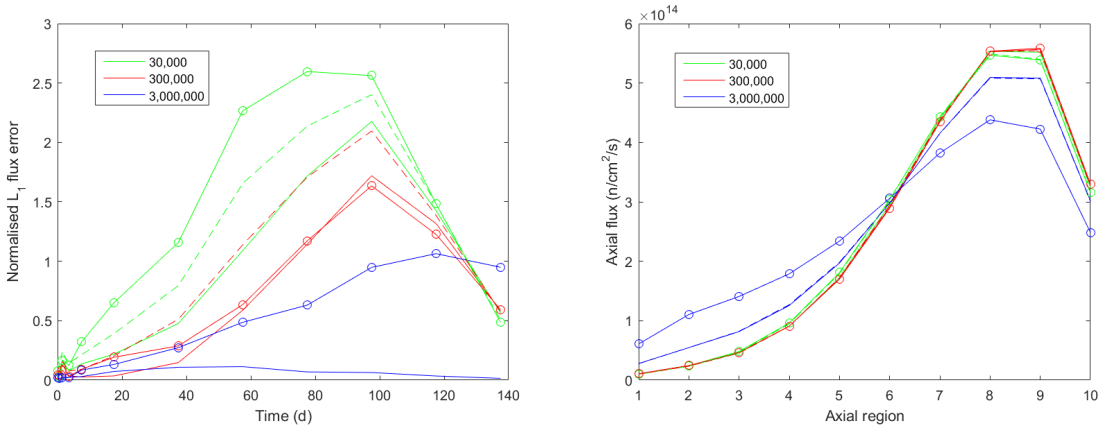


Fig. 2.8 Relative flux errors with time (left) and final flux profiles (right) when depleting a PWR pin with non-uniform coolant density using the explicit Euler scheme, with a 20 day maximum time-step, and varying the number of particles per generation while preserving the active particle histories.

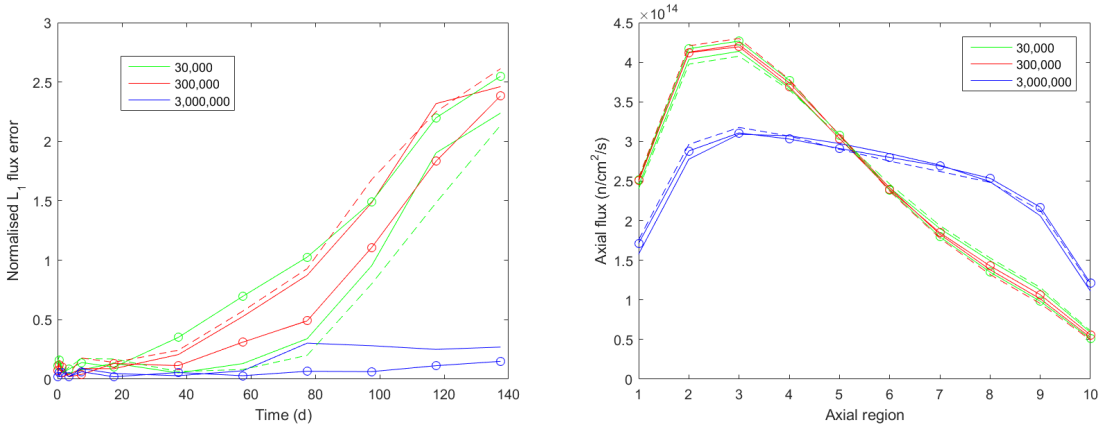


Fig. 2.9 Relative flux errors with time (left) and final flux profiles (right) when depleting a PWR pin with non-uniform coolant density using the PC scheme, with a 20 day maximum time-step, and varying the number of particles per generation while preserving the total particle histories.

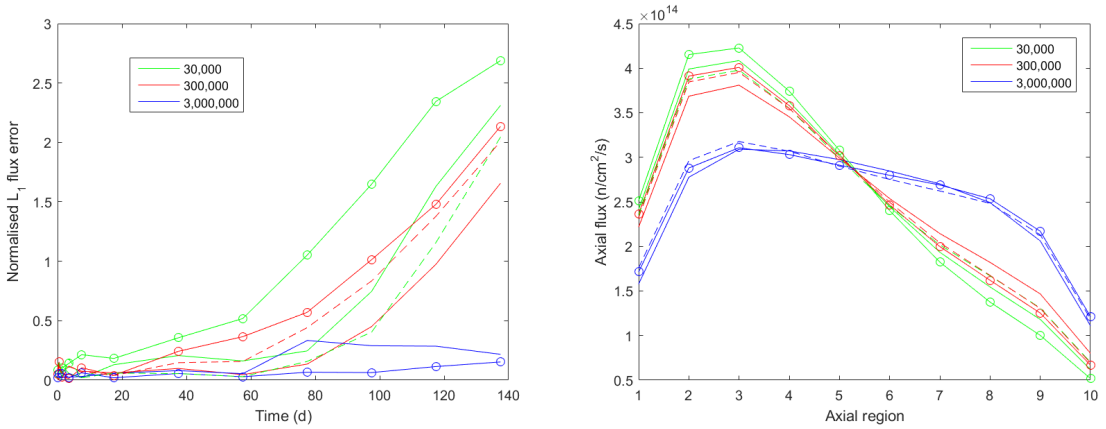


Fig. 2.10 Relative flux errors with time (left) and final flux profiles (right) when depleting a PWR pin with non-uniform coolant density using the PC scheme, with a 20 day maximum time-step, and varying the number of particles per generation while preserving the active particle histories.

For the Euler scheme, the error in the simulations using fewer particles initially diverges from the 3,000,000 particle cases by a large margin, showing that the solutions are markedly different. However, the errors ultimately begin to shrink back to zero. This appears only to be the case due to the reference solution, the dashed blue line in Figs. 2.7 and 2.8, beginning to converge towards the same solution as achieved by the simulations using fewer particles. Note the blue circled line in Figs. 2.7 and 2.8: its error begins to diverge but, in actuality, it is converging more slowly towards the 'unstable' state than is the reference solution, as can be seen from the plot of the final flux distributions.

This behaviour is significantly different as compared to the uniform coolant density case – previously using many particles per generation with the explicit Euler scheme managed to maintain a physically reasonable solution, whereas in these cases all Euler solutions transition markedly towards a peaked, non-physical solution, albeit slightly more slowly when using many particles per generation. This also holds when preserving only the number of active particles simulated. From Fig. 2.8, the simulations using fewer particles per generation fall into error more rapidly according to their L_1 error evolution.

More striking are the results from the PC scheme where, by the end of the simulations, two distinct sets of solutions exist. The simulations using fewer particles appear to have reached a distinctly peaked state, whereas the 3,000,000 cases obtain a much flatter profile. Within these two sets of solutions, all cases appear to be, qualitatively, quite consistent. This demonstrates that clustering can affect burn-up simulations significantly, even in cases which do not have a symmetry to be broken. This applied when preserving either total or active

histories, albeit, again, the error in the 300,000 particles per generation cases appears to grow slightly more slowly than the 30,000 particles per generation cases when preserving active histories.

Using 40 day maximum time-steps, the results for the explicit Euler are presented in Figs. 2.11 and 2.12, while the PC results are shown in Figs. 2.13 and 2.14. In this case, the behaviour of the Euler and PC schemes is nearly identical: the clustered simulations begin to diverge initially (while the time steps are still relatively short), before the 3,000,000 simulations begin to gravitate towards the very same solution, with all ultimately reaching the same state. Although, as with the uniform coolant case implying the instability of Euler scheme using 3,000,000 particles, these results confirm that this is equally the case for the PC scheme, subject to long time-steps and an asymmetric coolant profile. Hence, although clustering must be accounted for, it does not rule out the existence of true numerical instabilities, limiting time-step length or necessitating the use of implicit coupling schemes.

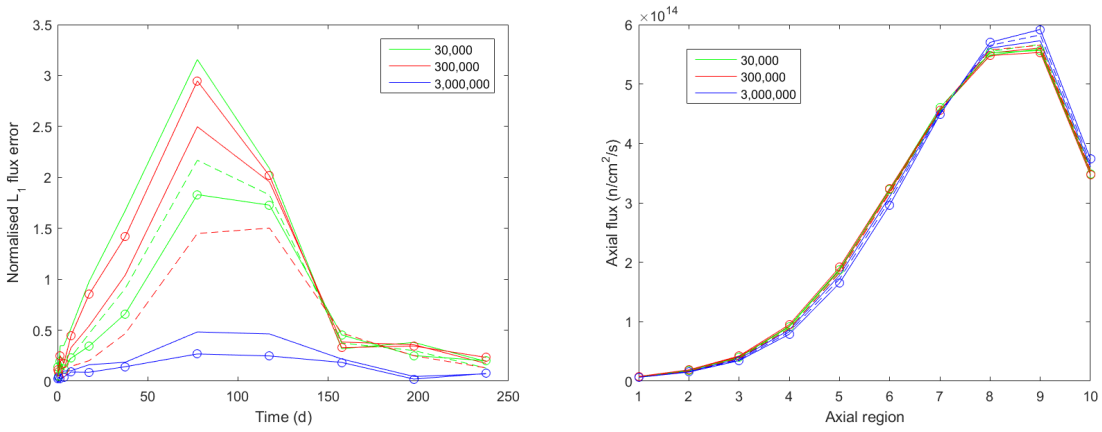


Fig. 2.11 Relative flux errors with time (left) and final flux profiles (right) when depleting a PWR pin with non-uniform coolant density using the explicit Euler scheme, with a 40 day maximum time-step, and varying the number of particles per generation while preserving the total particle histories.

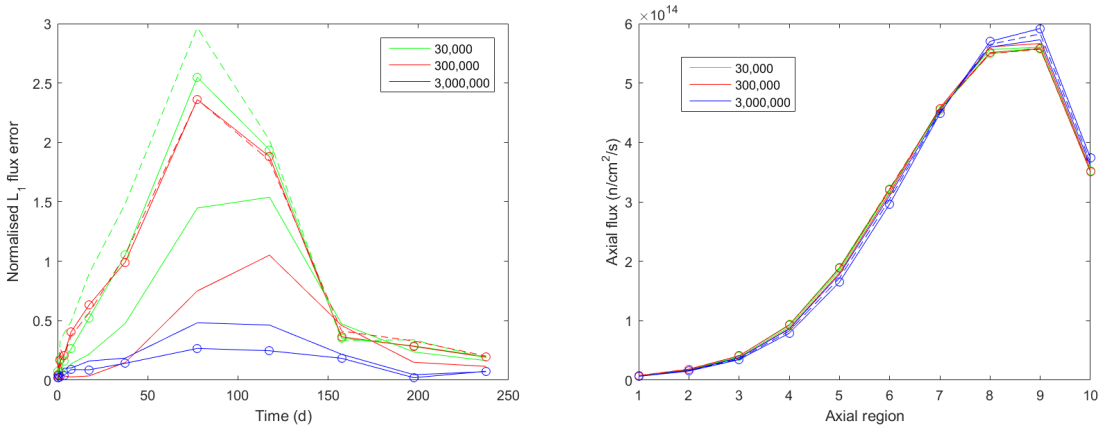


Fig. 2.12 Relative flux errors with time (left) and final flux profiles (right) when depleting a PWR pin with non-uniform coolant density using the explicit Euler scheme, with a 40 day maximum time-step, and varying the number of particles per generation while preserving the active particle histories.

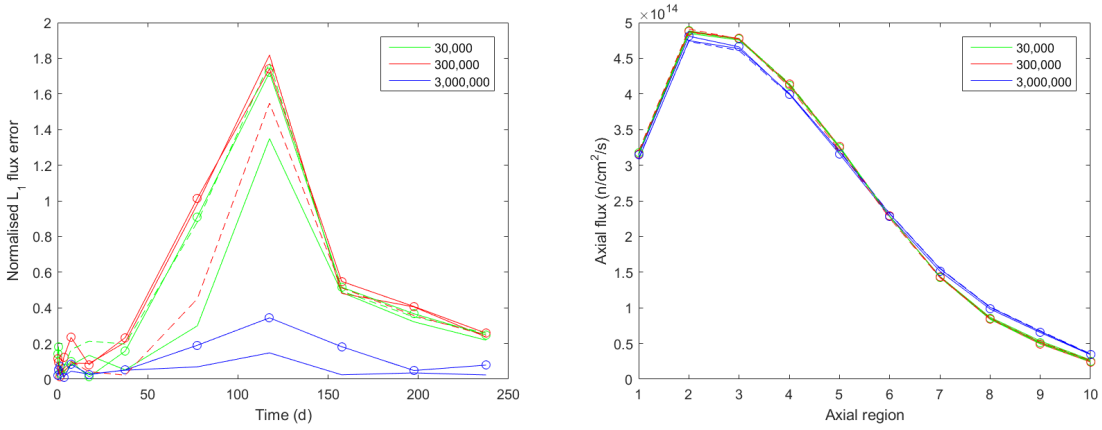


Fig. 2.13 Relative flux errors with time (left) and final flux profiles (right) when depleting a PWR pin with non-uniform coolant density using the PC scheme, with a 40 day maximum time-step, and varying the number of particles per generation while preserving the total particle histories.

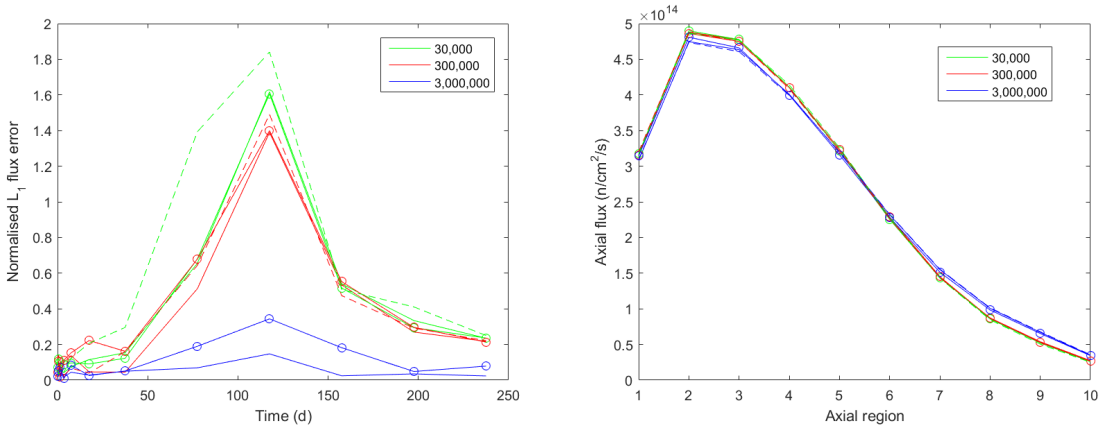


Fig. 2.14 Relative flux errors with time (left) and final flux profiles (right) when depleting a PWR pin with non-uniform coolant density using the PC scheme, with a 40 day maximum time-step, and varying the number of particles per generation while preserving the active particle histories.

2.4 Clustering and the simulation of xenon transients

Although most practical burn-up problems are concerned with reactor behaviour over days, months, and years, there is reason to consider shorter time-scales. The most obvious example would be when investigating a xenon transient. Unfortunately, a number of studies have demonstrated that similar difficulties arise here: both Isotalo et al. and Josey have shown that, considering a 4 m tall PWR pin with uniform coolant density and fully-reflective boundaries with short time-steps (15 minutes for Isotalo et al., unspecified by Josey, although apparently about 3 hours), an oscillatory solution also begins to develop [50, 51].

The beginning of these xenon oscillations occurs randomly, as can be seen from Fig. 1 in the paper by Isotalo et al., suggesting a stochastic element – this figure has been reproduced here as Fig. 2.15. Intriguingly, Isotalo et al. also consider a fuel pin with differences in enrichment along its length. The results for this are also shown in their Fig. 1. What is remarkable about the results presented are that a number of independent simulations agree on the timing of the flux oscillation – but one simulation does not, and is noticeably offset in its oscillation by about 3 hours, in spite of the very short (15 minute) time-step that was used. This implies that even problems which need not be concerned with statistics breaking symmetry are susceptible to erroneous results driven by stochastic phenomena – oscillations are expected to occur in different realisations of this case, but they should occur in phase with each other. Although Josey suggests that the phenomenon has some physical basis,

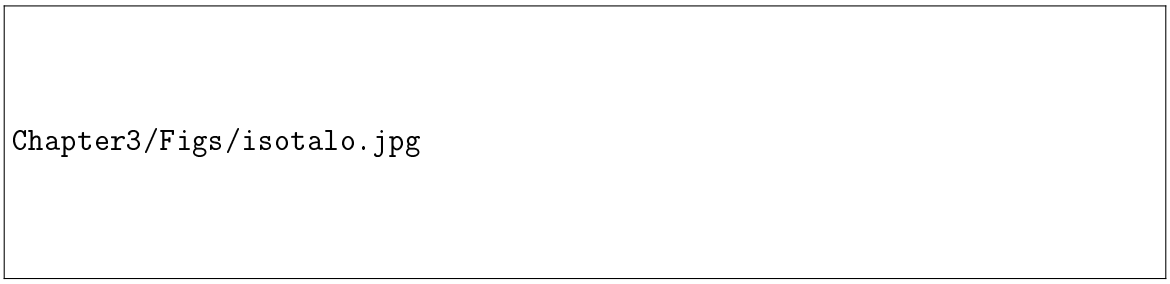


Fig. 2.15 Flux in a burnable region with 0 (left) and 0.1 (right) pp enrichment differences and 15 min steps without equilibrium xenon [50].

oscillations in a completely uniform problem can only be spurious – the oscillation can only be induced by a perturbation or an asymmetry not present in either the neutron transport equation or the Bateman equation. This suggestion may be motivated by Josey's application of the SIE scheme to the problem – this also produces an oscillatory result. There is the implication (based on Josey's Table 1.1) that 10 SIE iterations were used, although this is not explicitly stated.

Xenon oscillations do occur in real systems [6], but they are caused by, for example, control rod movements which are asymmetric. In this system the asymmetry is an artefact of the Monte Carlo transport solver – potentially some combination of limited numerical precision, stochastic noise, and correlation in the estimators. If an infinite number of neutrons were simulated each generation, the transport equation would be solved exactly (albeit still within the limits of numerical precision) and one would not so readily anticipate oscillations. However, with a limited number of neutrons each generation, Monte Carlo does not exactly solve the transport equation. On the other hand, if one imagined a different physical reality where neutrons were born, transported, and absorbed together in large packets such that reactors only contained on the order of thousands-to-millions of 'free' neutron packets at any point in time, the transport equation would not be sufficient to model the neutronics of such a reactor and one might anticipate purely stochastically-driven oscillations as a physical phenomenon.

Nevertheless, these results are concerning: it would appear that Monte Carlo simulations cannot reliably reproduce a xenon transient. This sentiment is echoed by Isotalo et al. who claim that it is 'impossible, not just unpractical[sic]' to remove spurious oscillations, highlighting the need for a stabilising scheme – enforcing xenon equilibrium in their case.

However, both of these studies used relatively few particles per generation: Isotalo et al. used 1000 neutrons over 5,000 active cycles with the PC scheme, while Josey used 128,000 neutrons over 2,500 active cycles with the explicit Euler scheme (12,800 neutrons

over 2,500 active cycles with the SIE scheme). As discussed above, clustering effects are likely to have contaminated the transport solutions. Hence, it is worth briefly investigating what effect clustering might have played in the results.

The investigations were done by taking the same model as used in Section 2.3 with a uniform coolant density. The only modifications made were to truncate the problem by removing the reflectors, and enforcing reflective boundaries, rather than vacuum. This problem is slightly shorter than that simulated by Isotalo et al. and Josey, at 3.66 m rather than 4 m, and also uses 10 rather than their 8 burnable regions. It is not imagined that either of these discrepancies affects the conclusions significantly.

The problem was burned with 1 hour time-steps up to 4 days of total burn-up using the explicit Euler scheme only. Three population settings were used: 3,000,000 particles per cycle with 50 active cycles and 100 inactive cycles, 30,000 particles per cycle with 5,000 active cycles and 10,000 inactive cycles, and 30,000 particles per cycle with 500 active cycles and 100 inactive cycles. The first two choices of settings were used to delineate the effects of clustering while preserving the total number of histories. The third set was used to demonstrate what might occur if 'common practice' is applied without regard for clustering. Given the flux should be uniform in all regions at all times, a similar error metric is used as in [26]:

$$\varepsilon = \sum_{i=1}^{10} \left| \frac{\phi_i}{\bar{\phi}} - \frac{1}{10} \right| \tag{2.6}$$

$$\bar{\phi} = \sum_{i=1}^{10} \phi_i \tag{2.7}$$

Here, as above, ϕ_i is the scalar flux in region i at a given point in time. This flux error is presented in Fig. 2.16. To provide some additional context, both the xenon density and flux in the left-most burnable region are shown in Fig. 2.17. The 'CP' used in the legends denotes the 'common practice' simulation settings.

The results show that the case using many neutrons per generation develops only a minor flux error, while the two simulations using fewer particles per generation (even with one preserving the total number of histories) both begin to deviate substantially over time, dipping only as the flux oscillates from one extreme to the other. In fact, simulating more histories with fewer particles per generation appears to produce non-physical results more quickly as compared to the common practice simulation. This may be entirely by chance, but it demonstrates that using more particle histories alone is no guarantee of increased fidelity. Indeed, far fewer histories were simulated here than by Josey: both the 3,000,000 and 30,000 particle cases tracked approximately 150,000,000 active histories while Josey followed 640,000,000 [51]. Even so, the 3,000,000 particles per generation case reproduces

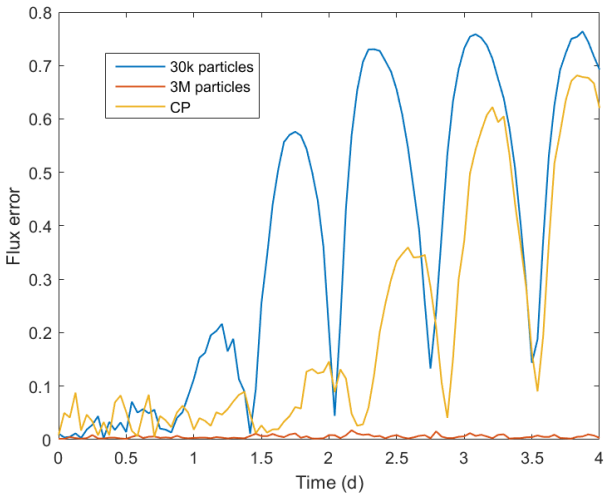


Fig. 2.16 Flux error when depleting a PWR pin with reflective boundaries using 1 hour time-steps with the explicit Euler scheme.

the results that would be expected: xenon increases initially before saturating at a constant level. Over longer times, one would expected to observe the flux increasing noticeably as fissile isotopes are consumed (in order to maintain a constant power), although this is not readily apparent over the quite brief period considered. Otherwise, the flux is apparently uniform across the geometry at all times. This appears to confirm that Monte Carlo neutron transport coupled with depletion can accurately follow a xenon transient, even when using the explicit Euler method.

Identical simulations were repeated for the same problem but featuring vacuum boundaries. The results were similar, albeit less dramatic: oscillations occur, but of a much smaller amplitude. As with the reflective case, simulating more particles per cycle appears to prevent their occurrence over the time-scales simulated.

2.5 Clustering and the gadolinium-bearing assembly

Section 1.6.1 highlighted the non-physical results obtained when simulating a mini-assembly containing gadolinium, subject to fully reflective boundary conditions. This problem, too, can be re-examined given an awareness of neutron clustering.

Recall that, in simulating this problem, xenon equilibrium was enforced, removing clustering-induced xenon oscillations as a potential cause of the instability. The mechanism by which gadolinium instability manifests was already described in Section 1.6.1: succinctly, a small error in the flux causes a positive feedback through increasing the local gadolinium

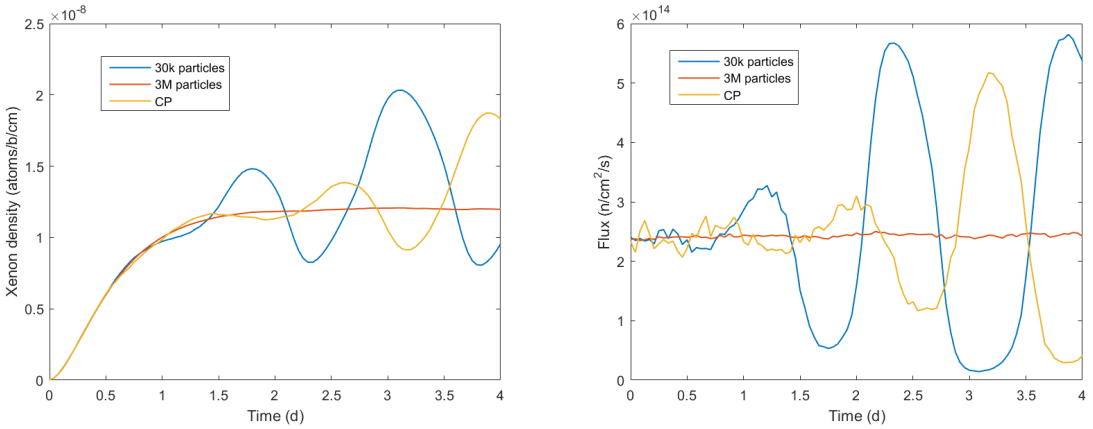


Fig. 2.17 Xenon density (left) and flux (right) in the left-most burnable region of a PWR pin with reflective boundaries when depleted with 1 hour time-steps using the explicit Euler scheme.

depletion and increasing the local flux. The suggestion in the current section is that clustering effects and correlation are largely responsible for asymmetries in the reaction rates, more so than statistical noise. Hence, minimising clustering should prevent or, at least, reduce the instability.

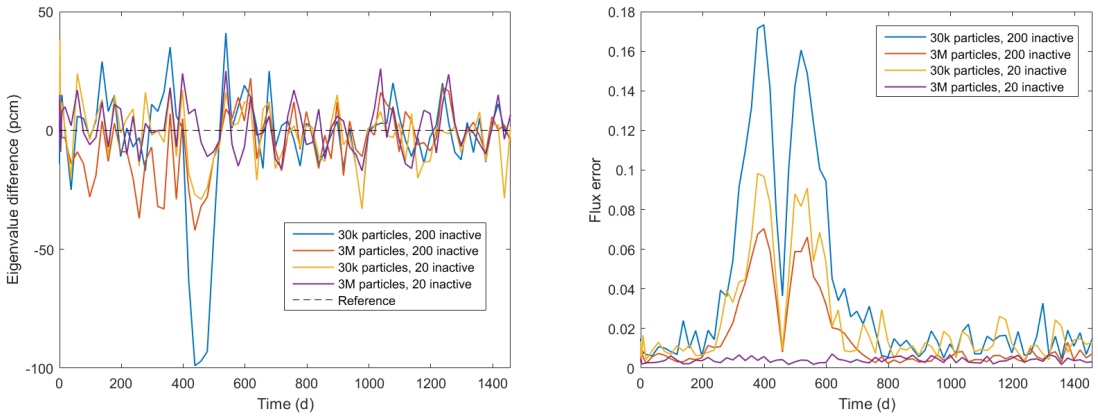
The mini-assembly problem from Section 1.6.1 was simulated with various particle settings. First were the settings show before (30,000 particles for 2,000 active cycles, 200 inactive), followed by a simulation using large generations settings while preserving the number of active histories: 3,000,000 particles for 20 active cycles and 200 inactive.

Alongside these two simulations were two which used the same active settings (with a large and small generation size) but reduced the number of inactive cycles substantially. These low inactive settings are justifiable in that the problem is axially uniform and therefore the axial component of the initial (uniform) source guess is correct. Hence, source convergence in this problem reduces to resolving the radial component which should be much easier than for the axial given the much smaller dimension (the assembly has a pitch of 3.78 cm and a height of 3.66 m).

Both the non-uniformity of these solutions (calculated using Eqs. (2.6) and (2.7) as done in Sections 1.6.1 and 2.4) and their deviation from the 2D eigenvalue are shown in Fig. 2.18. The maximum eigenvalue error reported by all simulations 9 pcm.

From the results it appears that reducing clustering and correlation effects do appear to lessen and eventually negate the observed instability – simulating either more particles or fewer particle generations tends to reduce the eventual eigenvalue error and nonuniformity of the flux. Apparently this problem is extremely sensitive to correlation-induced errors in the eigenvector given that there is a significant difference in behaviour between the

3,000,000 particle cases; an additional 180 inactive cycles is enough to trigger the instability. Furthermore, brief sensitivity investigations were conducted and significantly different results were produced by the 30,000 particle case using 20 inactive cycles – this simulation produced a much larger error than any of the others, in spite of using fewer inactive cycles than the other 30,000 particle case. This suggests that the problem is extremely sensitive to stochastic noise in the eigenvector. Hence, while this problem can be seemingly accurately resolved, it appears to be pathological without taking significant care.



(a) 2D/3D assembly eigenvalue comparison

(b) Non-uniformity of the 3D flux profile

Fig. 2.18 Stability investigation of the gadolinium-bearing mini-assembly with burn-up when using various particle settings.

2.6 Summary

This chapter began by noting that the problems for which burn-up instabilities have been reported are also likely to suffer from clustering and correlation affecting the Monte Carlo solver. Considering some previously proposed heuristics for minimising clustering, it was noted that most previous instability investigations used Monte Carlo settings which exacerbate these effects. For a PWR problem, similar to those investigated in other studies, using many particles per generation and fewer generations produced more stable results, using both the Euler and PC schemes. That said, the Euler scheme still developed an asymmetry, albeit more gradually, and stability still appears to depend substantially on the time-step and geometry. Clustering and correlation were then shown to have significantly affected previous reports of burn-up instabilities when taking extremely short time-steps; beforehand, these results called into question whether Monte Carlo neutronics was capable of accurately reproducing a xenon transient, with the presents results appearing to demonstrate that this is possible.

Finally, Isotalo's proposed 'gadolinium instability' was investigated and also shown to be substantially driven by correlation, even while enforcing xenon equilibrium.

Chapter 3

Stable methods for predictor-corrector schemes

The PC scheme discussed in previous chapters is familiar across many applications of numerical analysis [79]. Chapter 2 demonstrated that numerical instability may still be present for PC schemes if large time-steps are taken, even when accounting for neutron clustering – in fact, this can occur when using a 20 day time-step, depending on the problem in question. This chapter demonstrates that the corrector-step of the PC scheme and its variants can excite non-physical solutions and proposes two new corrector-step variations to improve its stability.

3.1 The persistence of instabilities

Chapter 2 demonstrated that neutron clustering has played a large role in the occurrence of non-physical results when coupling Monte Carlo neutronic and burn-up. However, it was also highlighted that, when taking large time-steps, non-physical results may yet be obtained, albeit not as immediately as when neutron clustering is a significant factor.

In fact, depending on the problem, even the 20 day time-steps used in the previous chapter will excite higher solution modes, provided enough time-steps are taken. The same PWR pin with vacuum boundaries and a uniform coolant density as used in the previous chapter was simulated in Serpent, again using 3,000,000 particles per generation, 60 active cycles and 10 inactive cycles, with source convergence accelerated by Serpent's in-built response matrix method using 1 source batch for 20 outer iterations [66]. The standard PC scheme was used with the same burn-up schedule with 20 day time-steps as done in the previous chapter. The only difference was that an additional 10 burn-up steps of 20 days in length

were performed. As a comparator, an identical simulation was run but which used a 60 day maximum time-step length – naturally, this is anticipated to be more unstable. Both sets of results are shown in Fig. 3.1, with the asymmetry calculated using Eq. (2.4) and the flux plotted at the final time-point.

From the graphs, both time-stepping schedules are unstable, although shorter time-steps delay the evolution of erroneous solution modes. Hence, while minimising clustering and correlation induced by population control has lessened the severity of non-physical behaviour and its speed of onset, numerical instability remains a problem, even when taking moderately-sized time-steps.

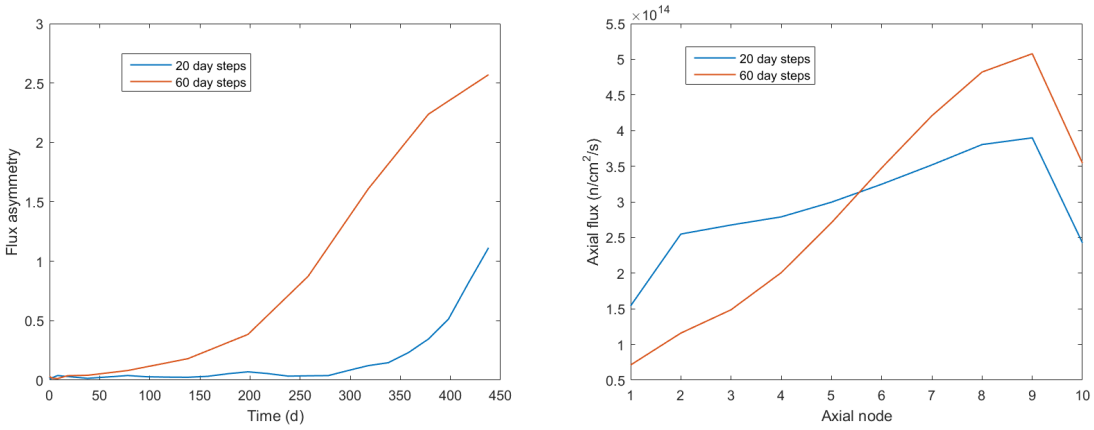


Fig. 3.1 Flux asymmetries with time (left) and final flux profiles (right) when depleting a PWR pin with uniform coolant density using the PC scheme, using a 20 day and 60 day maximum time-step.

3.2 The corrector step as a discrete non-linear map

The standard CE/LI PC scheme (without substeps) can be described with two equations:

$$N_{n+1} = \exp\left[\mathbf{A}(\psi_{\mathrm{BOS}})\Delta t\right] N_n \tag{3.1}$$

$$N_{n+1} = \exp\left[\frac{\mathbf{A}(\psi_{\mathrm{BOS}}) + \mathbf{A}(\psi_{\mathrm{EOS}})}{2}\Delta t\right] N_n \tag{3.2}$$

Here n is the time-step index, and each burn-up matrix, $\mathbf{A}$, is obtained from a transport solution, ψ. Eq. (3.1) is the predictor depletion calculation, while Eq. (3.2) is the corrector

calculation. Obtaining a transport solution, given a nuclide density field, can be written as:

$$\psi = \tau(N) \tag{3.3}$$

Here $\tau(\cdot)$ represents the action of a transport solver on the input nuclide density field. Note that, as discussed in Section 1.3, each material region in a coupled neutronics/depletion problem is independent of every other during the depletion calculation. Therefore, the burn-up matrix describing the entire problem is block diagonal and each region may be burned separately.

An alternative approach to the standard PC paradigm applied in reactor physics is to perform an iteration on the corrector step. Doing so and removing the BOS and EOS notation for brevity, with the latter replaced by an iteration index, gives:

$$N_{n+1}^{(0)} = \exp\left[\mathbf{A}(\psi)\Delta t\right] N_n \tag{3.4}$$

$$N_{n+1}^{(k+1)} = \exp\left[\frac{\mathbf{A}(\psi) + \mathbf{A}\left(\psi^{(k)}\right)}{2}\Delta t\right] N_n \tag{3.5}$$

Each corrector iteration still requires the evaluation of $\mathbf{A}(\psi^{(k)})$, making it relatively computationally expensive to use this variant of the PC scheme. In the literature, when using Monte Carlo as the transport solver, an iterated corrector step has only been found in two contexts: the first is where SI schemes are used, albeit, differing slightly due to the application of a variable relaxation factor. The second is a brief reference to its use in MC21 'to improve convergence of the [EOS nuclide] densities', without further discussion [37, 39]. The MONTEBURNS code [78] possesses an analogous iteration capability, albeit, applied to a slightly different PC paradigm (the CE/CM method by Isotalo's convention [46]). A version of the LE/LI scheme which iterates on the corrector step is shown in Algorithm 6.

One may write Eq. (3.5) to show explicit dependence upon the nuclide density from the previous iteration:

$$N_{n+1}^{(k+1)} = \exp\left[\frac{\mathbf{A}(\psi) + \mathbf{A}\left(\tau(N_{n+1}^{(k)})\right)}{2}\Delta t\right] N_n \tag{3.6}$$

This can be easily modified when taking n_s substeps during the corrector:

$$N_{n+1}^{(k+1)} = \prod_{s=1}^{n_\mathrm{s}} \left(\exp\left[\frac{w_{n,s}\mathbf{A}(\psi) + w_{n+1,s}\mathbf{A}\left(\tau(N_{n+1}^{(k)})\right)}{n_\mathrm{s}}\Delta t\right] \right) N_n \tag{3.7}$$

Input: $N_0, n_{\text{steps}}, \Delta t, n_{\text{ss}}, k_{\max}$
for $n = 0, \ldots, n_{\text{steps}} - 1$ **do**
  $\mathbf{A}_n \leftarrow \psi(N_n)$
  if $n > 0$ **then**
    $N \leftarrow N_n$
    for $s = 0, \ldots, n_{\text{ss}} - 1$ **do**
      $\mathbf{A} \leftarrow w_{n,s}\mathbf{A}_n + w_{n-1,s}\mathbf{A}_{n-1}$
      $N \leftarrow \exp\left[\mathbf{A}\frac{\Delta t_n}{n_{\text{ss}}}\right] N$
    end
    $N_{n+1}^{(0)} \leftarrow N$
  else
    $N_{n+1}^{(0)} \leftarrow \exp\left[\mathbf{A}_n^{\text{BOS}}\Delta t_n\right] N_n$
  end
  for $k = 0, \ldots, k_{\max} - 1$ **do**
    $\mathbf{A}_{n+1}^{(k)} \leftarrow \psi(N_{n+1}^{(k)})$
    $N \leftarrow N_n$
    for $s = 0, \ldots, n_{\text{ss}} - 1$ **do**
      $\mathbf{A} \leftarrow w_{n,s}\mathbf{A}_n + w_{n+1,s}\mathbf{A}_{n+1}^{(k)}$
      $N \leftarrow \exp\left[\mathbf{A}\frac{\Delta t_n}{n_{\text{ss}}}\right] N$
    end
    $N_{n+1}^{(k+1)} \leftarrow N$
  end
  $N_{n+1} \leftarrow N_{n+1}^{(k_{\max})}$
end

Algorithm 6: LE/LI substep method with corrector iteration.

Here, $w_{n,s}$ and $w_{n+1,s}$ are the weights for substep s of the BOS and EOS burn-up matrices, respectively. As the exponential terms in equation Eq. (3.7) are matrices, their multiplication must be ordered, with the $s = 1$ matrix the rightmost, and the $s = n_\mathrm{s}$ matrix the leftmost. The iterated corrector, like many other coupled physics problems, has the form of a non-linear map:

$$N_{n+1}^{(k+1)} = F\left(N_{n+1}^{(k)}\right) \tag{3.8}$$

Here F combines the action of the transport solver, τ, with that of the Bateman solver, more succinctly written as β, like so:

$$N_{n+1}^{(k+1)} = \beta\left(\Delta t, N_n, \psi, \tau\left(N_{n+1}^{(k)}\right)\right) = F\left(N_{n+1}^{(k)}\right) \tag{3.9}$$

The behaviour of non-linear maps is described by the theory of dynamical systems. In the context of depletion, on iteration, one would hope that the nuclide density vector would eventually converge to a constant vector (or fixed point) – within numerical precision and allowing for stochastic noise from the transport solver. Such a vector, N_{n+1}^{*}, would satisfy the equation:

$$N_{n+1}^{*} = F\left(N_{n+1}^{*}\right) \tag{3.10}$$

Less desirably, the nuclide density vector might diverge or oscillate endlessly, with both outcomes being obviously non-physical. Qualitative details of the map's behaviour can be found by considering its Jacobian matrix at the fixed point – that is, the matrix of derivatives of Eq. (3.8) with respect to the elements of its argument. This is denoted as:

$$J = \frac{\partial F\left(N_{n+1}\right)}{\partial N_{n+1}} \tag{3.11}$$

By Taylor expansion, this Jacobian allows for a local linearisation of the non-linear map about the fixed point as:

$$F\left(N_{n+1}^{(k)}\right) \approx F\left(N_{n+1}^{*}\right) + J \cdot \left(N_{n+1}^{(k)} - N_{n+1}^{*}\right) \tag{3.12}$$

provided $N_{n+1}^{(k)}$ is sufficiently close to the fixed point. The Jacobian's eigenvalue with the largest magnitude – the spectral radius, denoted $\rho(J)$ – will describe the behaviour of the discrete map over iterations. For discrete maps, if $\rho(J) < 1$, the iteration described by Eq. (3.8) will ultimately converge to the fixed point. Otherwise, if $\rho(J) > 1$, the iteration will diverge. In cases where $\rho(J) = 1$, higher order terms are required to determine convergence properties.

Table 3.1 Burn-up schedule with a maximum time-step of 60 days.

Time-point	Time (d)	Burn-up (MWd/kg)
1	0	0
2	1	0.034
3	3	0.10
4	8	0.27
5	18	0.61
6	38	1.3
7	78	2.6
8	138	4.7
9	198	6.7
10	258	8.7
11	318	11
12	378	13
13	438	15

3.2.1 Numerical study on corrector-step iteration

If a coupled transport-depletion problem is stable, it should converge to a fixed nuclide density field and transport solution when iterating the corrector step. For a symmetric problem, it should also maintain its symmetry on iteration. Using the uniform density PWR pin described in Section 2.3, this is investigated for the standard PC method (CE/LI without substeps), as well as the CE/LI and LE/LI methods with 10 substeps. The calculations are performed using maximum time-steps 60 days, with 1 (as standard) to 3 corrector iterations. The time-stepping schedule and burn-up are shown in Table 3.1 for clarity.

Serpent v2.1.31 is used as the transport solver. However, it has been modified to output depletion information such as one-group microscopic cross-sections, fluxes, fission yields, decay constants, and the identity of each nuclide that Serpent would normally track. These are used in an external depletion engine, written in MATLAB and using a CRAM solver, to allow for more variations in the depletion scheme than Serpent has currently implemented. Although all of the methods mentioned in the above paragraph are included in Serpent, their iterated versions are not. With some experimentation in running the short time-step problem presented in Section 2.4, 2 million particles per generation were found adequate to avoid contamination of the results due to clustering effects. As in Section 3.1, Serpent's response matrix acceleration method was used to reduce the number of inactive cycles applied down to 10, while 80 active cycles were used. The response matrix settings used 1 source batch for 20 outer iterations. The results are shown in Fig. 3.2.

First, without iteration, it seems that all time-stepping schemes are unstable for the given time-step length. Furthermore, the LE/LI scheme in particular seems marginally more

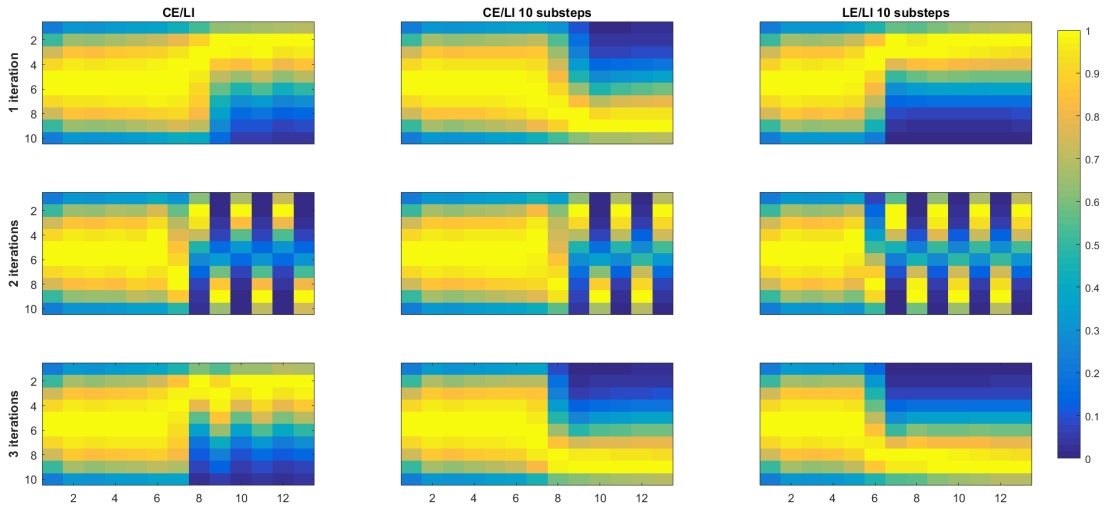


Fig. 3.2 Flux profile over time for a PWR fuel pin with axially vacuum boundaries using three different burn-up schemes and varying the number of corrector iterations. The horizontal axes on each graph are the burn-up time-points given in Table 3.1, while the vertical axes are the flux values at a given axial node from 1 to 10, with the fluxes at a given time-point normalised by the maximum value.

unstable than the others in that the flux begins to tilt more quickly and more extremely – on the sixth step, rather than the eighth for both CE/LI variations. It may be the case that the LE predictor more readily excites higher solution modes than the CE; this was hinted at by Isotalo et al. [50], who found that, even when enforcing xenon equilibrium, oscillations occur with the LE scheme when taking relatively short time-steps of 30 days. It can also be seen from the results that, for the problem considered, the corrector step is numerically unstable: when more iterations are carried out, an incorrect, asymmetric solution is obtained more rapidly than otherwise, typically making its appearance one or two steps earlier than when using only a single iteration. This suggests that modifying the corrector step may improve stability.

3.3 Relaxation

Many non-linear systems in numerical methods are unstable, i.e., have $\rho(J) > 1$. One of the most commonly employed techniques to attain a stable solution is 'relaxation', commonly used in coupling Monte Carlo neutronics with thermal-hydraulics, as thoroughly discussed by Gill et al. [36]. Applying relaxation to the corrector step's iterated discrete map gives:

$$N_{n+1}^{(k+1)} = \alpha F\left(N_{n+1}^{(k)}\right) + (1-\alpha)N_{n+1}^{(k)} \tag{3.13}$$

Here, $0 < \alpha < 1$, where α is known as the 'relaxation factor'. Practically, relaxation means that each evaluation of the discrete map only partially contributes to the next solution with weight α, the remainder being contributed by the previous solution with a weight of $1 - \alpha$. The greater the value of α, the faster the convergence, but at the risk of being unstable if α is too large. In the context of depletion, this means that performing a relaxation on the corrector step may affect solution accuracy, even while improving stability. An aggressive relaxation factor will place emphasis on the nuclide density vector from the predictor-step extrapolation, with a relatively minor contribution from the more accurate corrector interpolation. With sufficient iterations, convergence to the correct solution vector should occur within the bounds of stochastic noise but will be slowed if α is too small. This must be accounted for when choosing the relaxation factor and number of iterations.

This scheme requires only a slight modification to the standard PC variants and is shown in Algorithm 7 for the LE/LI scheme.

It was mentioned in Section 1.6.2 that SI schemes are relaxation schemes, albeit with a variable relaxation factor, $\alpha_k = \frac{1}{k+1}$, such that they may be written [27]:

$$N_{n+1}^{(k+1)} = \alpha_k F\left(N_{n+1}^{(k)}\right) + (1 - \alpha_k) N_{n+1}^{(k)} \tag{3.14}$$

The first iteration essentially does not apply a relaxation, while α_k decreases monotonically thereafter. As discussed previously, SI schemes are often slow to converge, usually requiring on the order of 10 iterations to provide a stable solution. Even when this is not the case, it was demonstrated in Chapter 1 that determining how many SI iterations to perform for a given computational budget is not straightforward. The stochastic approximation was initially devised to ensure that the fixed point of noisy functions could be located without any guarantee of efficiency or optimality [85]. However, at least for the burn-up problems investigated in this thesis, Monte Carlo noise is quite small – on the order of 0.1% error in the one-group flux for a single burnable region, although this is an under-estimate of the true error [42]. When using relatively small neutron populations per generation, given the effects of neutron clustering on depletion, the stochastic approximation's averaging across iterations may be necessary to negate biases from the clustering-affected transport solutions. The resulting small relaxation factor should also prevent numerical instability. However, if in such cases this averaging is necessary to achieve stable solutions, when the effects of clustering are minimised by judicious choice of Monte Carlo settings, it should be possible to improve upon SI schemes. In particular, there would no longer be a need for a monotonically decreasing relaxation factor – applying some fixed value may achieve a stable solution using

Input: $N_0, n_{\text{steps}}, \Delta t, n_{\text{ss}}, k_{\max}, \alpha$
for $n = 0, \ldots, n_{\text{steps}} - 1$ **do**
  $\mathbf{A}_n \leftarrow \psi(N_n)$
  if $n > 0$ **then**
    $N \leftarrow N_n$
    for $s = 0, \ldots, n_{\text{ss}} - 1$ **do**
      $\mathbf{A} \leftarrow w_{n,s}\mathbf{A}_n + w_{n-1,s}\mathbf{A}_{n-1}$
      $N \leftarrow \exp\left[\mathbf{A}\frac{\Delta t_n}{n_{\text{ss}}}\right]N$
    end
    $N_{n+1}^{(0)} \leftarrow N$
  else
    $N_{n+1}^{(0)} \leftarrow \exp\left[\mathbf{A}_n \Delta t_n\right]N_n$
  end
  for $k = 0, \ldots, k_{\max} - 1$ **do**
    $\mathbf{A}_{n+1}^{(k)} \leftarrow \psi(N_{n+1}^{(k)})$
    $N \leftarrow N_n$
    for $s = 0, \ldots, n_{\text{ss}} - 1$ **do**
      $\mathbf{A} \leftarrow w_{n,s}\mathbf{A}_n + w_{n+1,s}\mathbf{A}_{n+1}^{(k)}$
      $N \leftarrow \exp\left[\mathbf{A}\frac{\Delta t_n}{n_{\text{ss}}}\right]N$
    end
    $N_{n+1}^{(k+1)} \leftarrow \alpha N + (1-\alpha)N_{n+1}^{(k)}$
  end
  $N_{n+1} \leftarrow N_{n+1}^{(k_{\max})}$
end

Algorithm 7: LE/LI substep method with corrector iteration and relaxation

fewer iterations, minimising computational expense. The remainder of this section will evaluate this alternative implicit scheme.

3.3.1 Numerical study on relaxed corrector-step iteration

To investigate the viability of a naive relaxation to achieve a stable solution, precisely the same experiments as described in Section 3.2.1 will be performed using a relaxed version of each of the schemes mentioned. The relaxation factor applied is varied from 0.3 to 0.6 with both the unrelaxed ($\alpha = 1$) and SI ($\alpha = \frac{1}{k+1}$) results included for comparison.

First, the CE/LI scheme without substeps is presented in Fig. 3.3. From the results, with a fixed, relatively low relaxation factor of 0.3 or 0.4, the CE/LI remains quite symmetric, becoming more so with additional corrector iterations. On the other hand, there is significant growth in the flux asymmetry when using a relaxation factor of either 0.5 or 0.6. While iteration ultimately damps the asymmetry growth with a relaxation of 0.5, it remains significant with a relaxation of 0.6. As the SI-scheme starts with relatively mild relaxation, it requires additional iterations to achieve a physical solution as opposed to simply relaxing with a factor of 0.3 or 0.4 which remain stable with a single iteration.

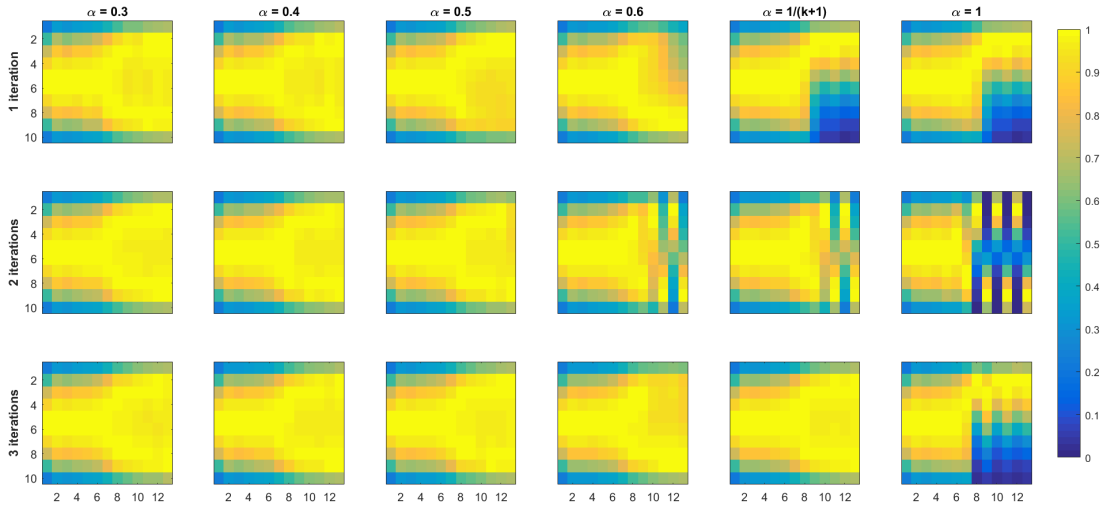


Fig. 3.3 Flux profile over time for a PWR fuel pin with axially vacuum boundaries using the CE/LI scheme with no substeps and varying the number of corrector iterations and relaxation factor. The horizontal axes on each graph are the burn-up time-points given in Table 3.1, while the vertical axes are the flux values at a given axial node from 1 to 10, with the fluxes at a given time-point normalised by the maximum value.

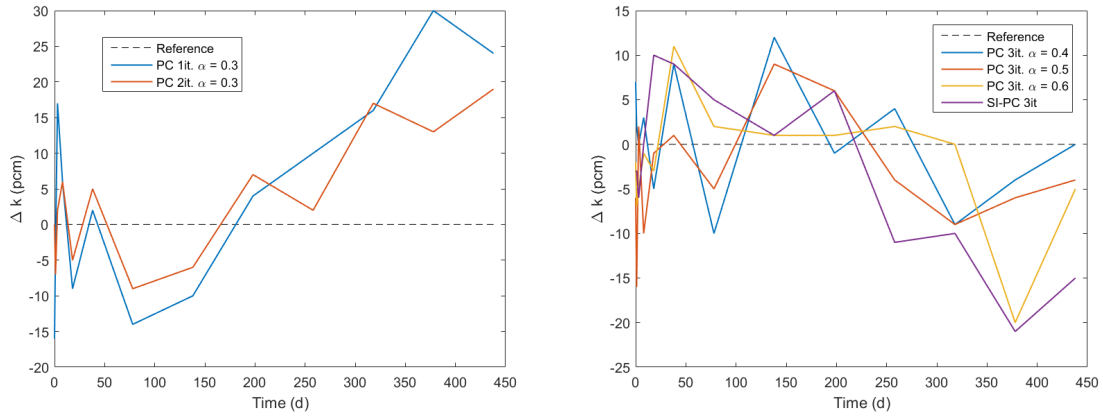


Fig. 3.4 Differences in the eigenvalue with burn-up for the relaxed CE/LI scheme without substeps when varying the number of iterations (left) and relaxation factor (right).

Regarding how accuracy may be degraded through relaxation, it is difficult to establish definitively without a reference. The simulation using a relaxation factor of 0.3 for 3 iterations will be taken as the reference case given that it is stable and has undergone a number of iterations – therefore, at each time-step, it should have approached the fixed-point during the corrector-step. Arguably, the relaxation factor of 0.4 might also be a suitable reference given it does not display any obvious signs of instability and, given the larger relaxation factor, may have converged more closely to the fixed-point than the 0.3 case. Two plots are shown in Fig. 3.4: the first compares the eigenvalue difference for a relaxation factor of 0.3 when varying the number of iterations, the second compares the eigenvalue difference for 3 iterations when using various relaxation factors. In all simulations, the maximum uncertainty in the eigenvalue was 5 pcm, with most at about 3.5 pcm.

Considering the eigenvalue error when fixing the relaxation at 0.3 and varying the number of iterations, there is an error accumulation over time, albeit a small one. Intriguingly, however, the maximum eigenvalue error is larger than the error from using a relaxation factor of 0.6 and three iterations which returned a significantly more obviously incorrect eigenvector than any simulation using a relaxation of 0.3. Without iteration or relaxation, the eigenvalue error reaches about 900 pcm – by comparison each of these modified schemes are quite accurate.

It should be emphasised in this section that the stable flux profiles, towards the end of the burn-up schedule, become quite flat such that Monte Carlo noise may appear to be substantial in the heatmaps presented. This may be emphasised by the presentation style used here with the flux at a given time-point normalised to the maximum value at the same time-point. To illustrate, the flux profiles without normalisation are shown in Fig. 3.5 for a single corrector iteration using relaxation factors of 0.3, 0.5, and 1. From the figure one can see that whatever

noise may be apparent in the flux profiles of Fig. 3.3 is slight in comparison to the unstable flux tilt that can manifest.

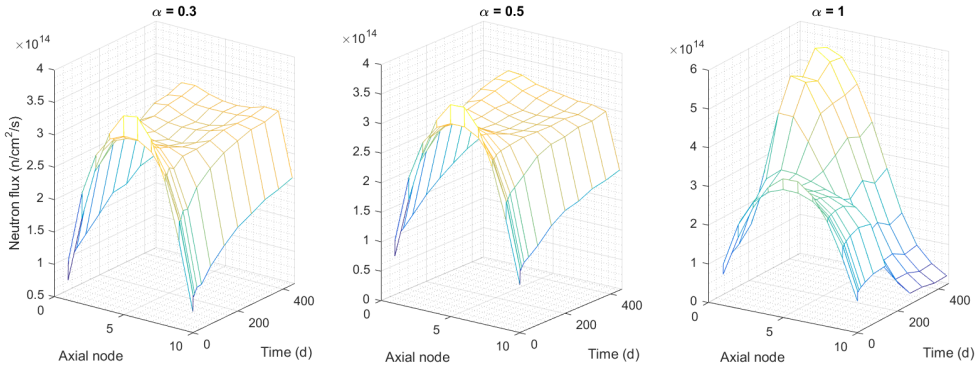


Fig. 3.5 Flux profiles over time for a PWR fuel pin with axially vacuum boundaries using the CE/LI scheme with no substeps, one corrector iteration and a variable relaxation factor.

The flux profile results for the CE/LI scheme using 10 substeps are shown in Fig. 3.6, while the corresponding eigenvalue errors are shown in Fig. 3.7. As compared to Fig. 3.3, it is apparent that adding substeps to the corrector results in a more unstable problem given that more aggressive relaxation appears necessary to produce solutions which remain symmetric with time.

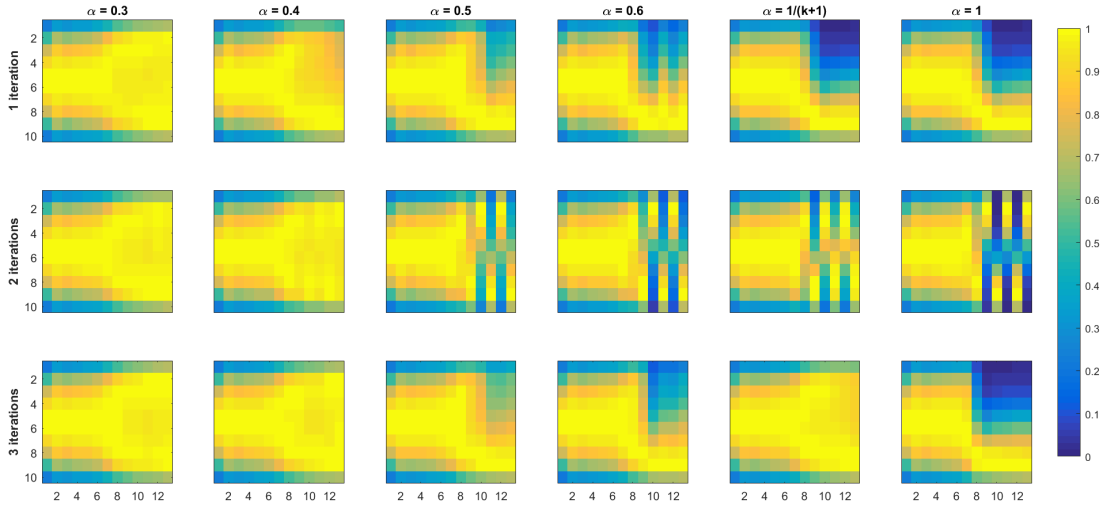


Fig. 3.6 Flux profile over time for a PWR fuel pin with axially vacuum boundaries using the CE/LI scheme with 10 substeps and varying the number of corrector iterations and relaxation factor. The horizontal axes on each graph are the burn-up time-points given in Table 3.1, while the vertical axes are the flux values at a given axial node from 1 to 10, with the fluxes at a given time-point normalised by the maximum value.

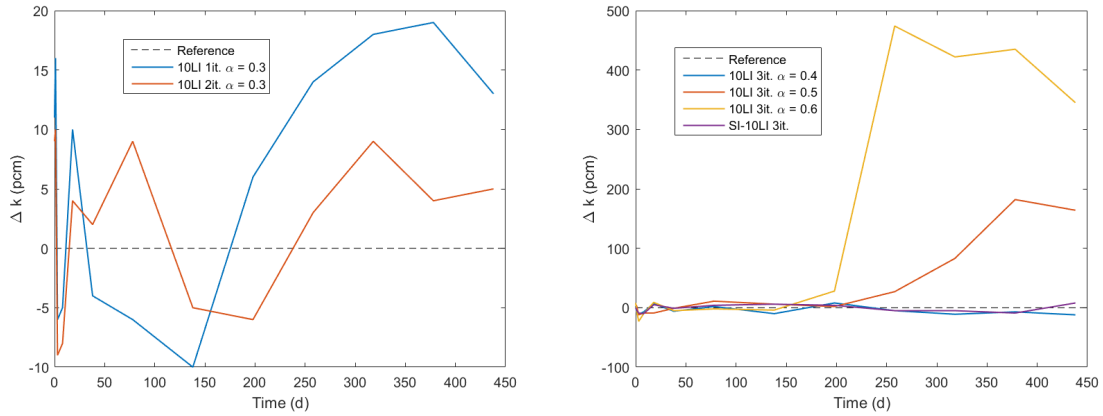


Fig. 3.7 Differences in the eigenvalue with burn-up for the relaxed CE/LI scheme with 10 substeps when varying the number of iterations (left) and relaxation factor (right).

Regarding the eigenvalue errors, all simulations using a relaxation factor of 0.3 show relatively good agreement, accounting for uncertainty. However, when varying the relaxation factor with 3 iterations, significant disparities emerge – both the 0.5 and 0.6 simulations unsurprisingly diverge in their eigenvalue. The SI relaxation scheme, on the other hand, shows good accuracy in the eigenvalue, in spite of the mild flux tilt which it develops. Without relaxation or iteration, the error in the eigenvalue reaches approximately 2,200 pcm.

Finally, the results for the LE/LI scheme with 10 substeps are presented in Figs. 3.8 and 3.9, both showing a significant tendency towards instability. In particular, for various relaxation factors with only a single corrector iteration, comparatively strange instability behaviour is observed compared to the predictable one-sided tilt that previous schemes displayed. This is presumably due to the effect of the LE extrapolation being relatively more unstable as compared to the CE extrapolation shown previously. For the LE/LI scheme, it seems that a stable result necessitates at least two iterations with a relaxation factor of 0.3 or three iterations with a relaxation factor of 0.4. Both of these settings provide a more symmetric solution than their SI-equivalent with the same number of iterations.

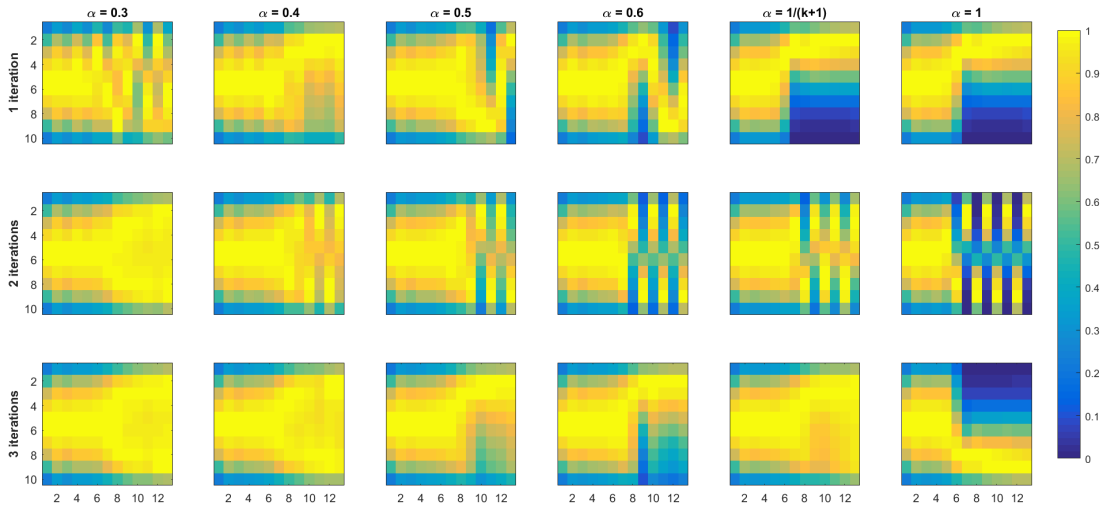


Fig. 3.8 Flux profile over time for a PWR fuel pin with axially vacuum boundaries using the LE/LI scheme with 10 substeps and varying the number of corrector iterations and relaxation factor. The horizontal axes on each graph are the burn-up time-points given in Table 3.1, while the vertical axes are the flux values at a given axial node from 1 to 10, with the fluxes at a given time-point normalised by the maximum value.

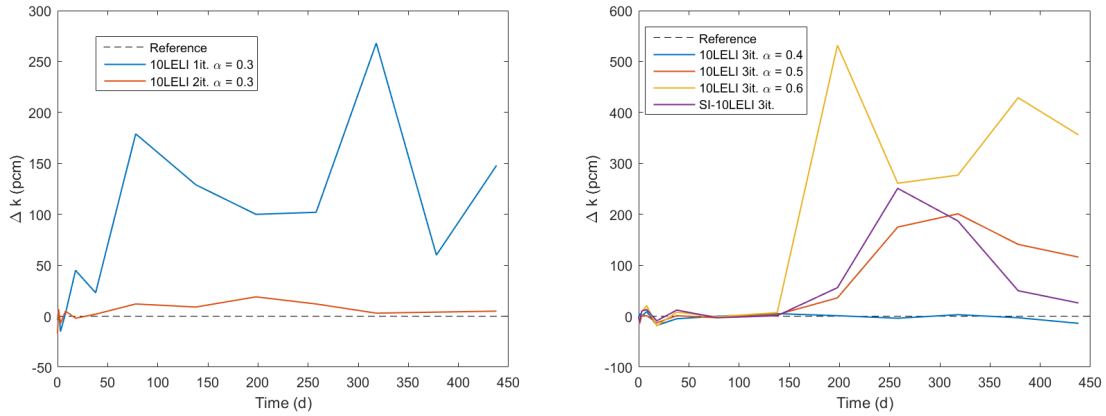


Fig. 3.9 Differences in the eigenvalue with burn-up for the relaxed LE/LI scheme with 10 substeps when varying the number of iterations (left) and relaxation factor (right).

From Fig. 3.9, the LE/LI scheme also produces more erroneous eigenvalues if not stabilised – as above, while two or more iterations with a relaxation factor of 0.3 seem to agree reasonably well, a single iteration returns an extremely biased eigenvalue. Furthermore, while relaxation factors of 0.3 and 0.4 show good eigenvalue agreement with three iterations, all other schemes produce biased results to various extents, even the SI-scheme.

In all cases examined here, a relaxation factor of 0.3 with either two or three corrector iterations seems sufficient to obtain realistic, stable results. However, the SI-scheme has previously been demonstrated to perform well when the statistics applied to each corrector transport calculation are reduced. To clarify, instead of running ten corrector solutions each following one billion particle histories, often with SI-schemes one runs the same number of corrector calculations but with a tenth of the histories to achieve a similar running time as compared to a standard PC scheme. As discussed in Section 1.6.2, one can often achieve the same stabilising effect at reduced computational expense. It is worthwhile investigating whether the fixed relaxation scheme possesses the same property. To do so, the same problem as above was simulated using the LE/LI scheme (as it was the most challenging scheme). Both the SI-scheme and the fixed relaxation scheme (for $\alpha = 0.3$) were used with three corrector iterations. The same statistics as above were used for each predictor transport simulation (2,000,000 particles for 80 active cycles, and 10 inactive with the response matrix solver). During each corrector solution, the number of active cycles was divided by three such that 26 cycles were used. In principle, one could also use fission source passing between transport solutions to reduce the number of inactive cycles, although this was not feasible here given the external depletion coupling used. That said, it is possible that fission source passing may also entail some adverse affects due to correlation. The resulting flux profiles for these two simulations are shown in Fig. 3.10, while the eigenvalue error against the three iteration LE/LI simulation with a relaxation factor of 0.3 and using full statistics is shown in Fig. 3.11.

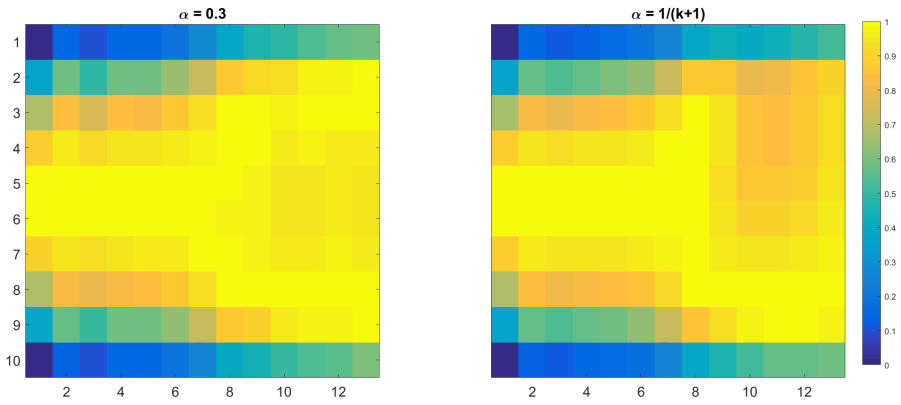


Fig. 3.10 Flux profile over time for a PWR fuel pin with axially vacuum boundaries using the LE/LI scheme with 10 substeps, three corrector iterations, and varying the relaxation scheme, while dividing the number of active histories during the predictor equally across corrector iterations. The horizontal axes on each graph are the burn-up time-points given in Table 3.1, while the vertical axes are the flux values at a given axial node from 1 to 10, with the fluxes at a given time-point normalised by the maximum value.

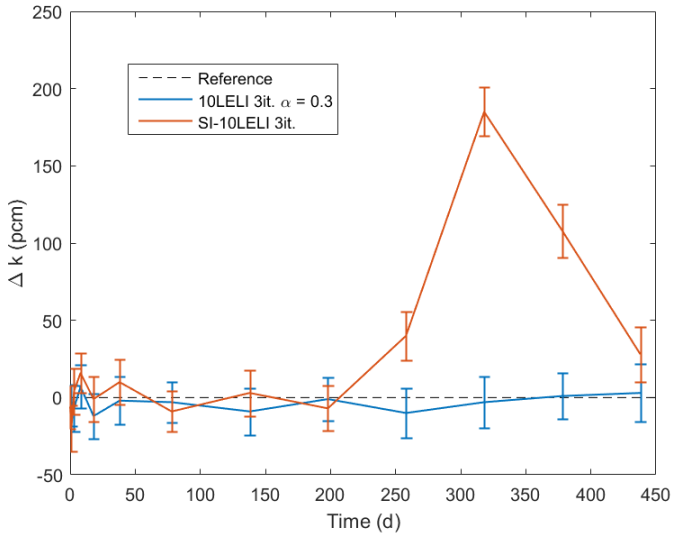


Fig. 3.11 Differences in the eigenvalue with burn-up for the relaxed LE/LI scheme with 10 substeps, three iterations and two relaxation schemes which use the same number of active histories across all corrector transport calculations as during the predictor.

In both figures, each scheme retains much the same behaviour as compared to their full statistics versions shown above – the fixed relaxation scheme remains symmetric, with an eigenvalue differing from the previous calculation within statistical uncertainty, whereas the SI-scheme similarly becomes asymmetric and accumulates a substantial eigenvalue error after about 150 days.

It is worth emphasising that reducing the number of histories per corrector should not be done recklessly. An extreme reduction may induce substantial statistical noise which might impact stabilising properties. Furthermore, as highlighted by Dumonteil & Diop [31], due to the non-linear coupling of Monte Carlo neutronics and depletion, a bias is induced in the matrix exponential. This bias is proportional to $\frac{1}{n}$, where n is the number of particle histories contributing to the depletion tallies. Hence, performing multiple corrector steps with reduced histories will increase this bias during each corrector step which may also adversely impact numerical stability.

Finally, one should investigate the extent to which this fixed relaxation scheme relies on accounting for neutron clustering effects. This can be straightforwardly done by reducing the number of particles per cycle while maintaining the same number of active histories. Hence, for the LE/LI scheme with 10 substeps, 3 corrector iterations, and relaxing with both a fixed factor of 0.3 and the stochastic approximation scheme, 200,000 particles per cycle are used over 800 cycles with 200 inactive cycles, applied to the same PWR pin problem as above. Response matrix acceleration would be used were it not for the original paper on response

matrix acceleration reporting convergence problems when simulating too few particles in a large geometry [66]. The results for both of these schemes are shown in Fig. 3.12 and compared against the LE/LI scheme with a relaxation factor of 0.3, using the same particle settings as shown previously.

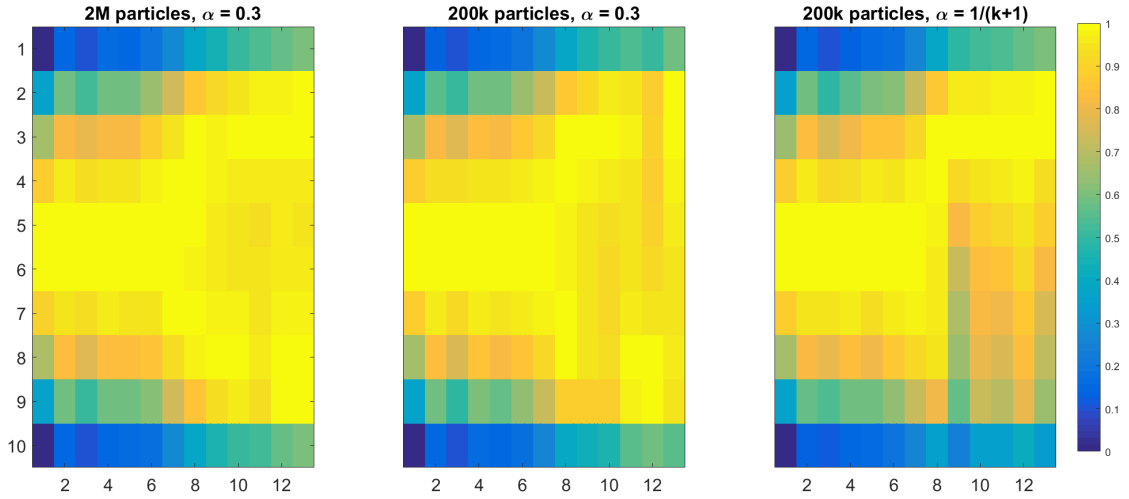


Fig. 3.12 Flux profile over time for a PWR fuel pin with axially vacuum boundaries using the LE/LI scheme with 10 substeps, three corrector iterations, and varying the relaxation scheme, while also varying the particle generation size. The horizontal axes on each graph are the burn-up time-points given in Table 3.1, while the vertical axes are the flux values at a given axial node from 1 to 10, with the fluxes at a given time-point normalised by the maximum value.

While the particle population size does obviously affect the stability of the fixed relaxation scheme, it appears to affect schemes using the stochastic approximation similarly adversely.

3.4 Xenon equilibrium with an iterated corrector-step

As discussed in Chapter 1, enforcing xenon equilibrium is a common method to ensure stability in reactor analysis problems and is a common feature of Monte Carlo codes with integrated burn-up capabilities. However, Isotalo et al. warn and the work of Densmore et al. implies that the method is not fool-proof, especially when taking over-long time-steps [24, 50]. In the author's experience, enforcing xenon equilibrium greatly improves stability but not unconditionally. Hence, this section will investigate the stability improvements gained by enforcing xenon equilibrium.

This section only considers the LE/LI scheme with 10 substeps as the previous results have shown it to be the most troublesome of the schemes considered. Furthermore, preliminary

Table 3.2 Burn-up schedule with a maximum time-step of 60 days when enforcing xenon equilibrium.

Time-point	Time (d)	Burn-up (MWd/kg)
1	0	0
2	1	0.034
3	3	0.10
4	8	0.27
5	18	0.61
6	38	1.3
7	78	2.6
8	138	4.7
9	198	6.7
10	258	8.7
11	318	11
12	378	13
13	438	15
14	498	17
15	558	19
16	618	21
17	678	23
18	738	25
19	798	27

investigations showed the outcomes of identical similar situations to differ significantly when using different random number seeds; as a result, each case investigated is simulated 5 times, each with a different seed. As done at the conclusion of the previous section, where the corrector step is iterated, the number of active cycles is divided by the number of iterations. A brief sensitivity investigation with two corrector iterations showed a non-physical tilt developing more rapidly when the corrector step maintained full statistics. Therefore, it is presumed that the results are relatively insensitive to dividing the active histories across multiple corrector steps. Finally, a larger number of time-steps are taken to further develop the instability behaviour. The burn-up schedule is provided in Table 3.2.

The flux profiles across the pin are shown in Fig. 3.13. Strikingly, large oscillations only appear to occur with even numbers of corrector iterations, consistently across all simulations. The magnitude of the non-physical oscillations vary between instances, although they do occur reliably. On the other hand, some mild asymmetry develops in some instances where only one or three corrector iterations are used, albeit, not so noticeably or reliably as in cases with the even-corrector oscillations. The underlying mechanism behind the oscillations (or lack thereof) is not as obvious as before – due to enforcing xenon equilibrium, it can no longer be attributed to a ‘numerical xenon oscillation’. Some explanation might be provided

by the results of Chapter 4 where it is shown that PC schemes with different numbers of corrector iterations can possess substantially different stability properties.

Non-physical behaviour occurring with an even number of corrector iterations might be considered an academic concern – when enforcing xenon equilibrium there is no need to perform additional corrector iterations to ensure stability, and few codes offer the option. However, where it might become inconvenient for a practising reactor physicist is when coupling neutronics, depletion, and other physical phenomena, such as thermal-hydraulics. One conceivable coupling scheme might involve, during the EOS, iteration between neutronics, thermal-hydraulics, and depletion to mutually converge each field. This would entail repeated corrector iterations which may result in non-physical solutions. Therefore, it is worth determining whether this behaviour can be easily averted. Following from the previous section, this is attempted by applying a relaxation factor to the corrector step. In doing so, all of the previous calculation are repeated but with a relaxation factor of $\alpha = 0.5$. This factor was chosen under the presumption that enforcing xenon equilibrium should provide some stabilising effect, and thus a harsh relaxation factor of $\alpha = 0.3$ might prove unnecessary. The results are shown in Fig. 3.14. An additional set of results are included as the bottom row where xenon equilibrium is not enforced; these follow the 3 corrector iterations and $\alpha = 0.3$ calculations used in the previous section. In each of these cases, the results appear in good agreement, regardless of the number of iterations. Perhaps the most obvious difference across the simulations is between those which do and do not enforce xenon equilibrium during the initial (short) time-steps: the former set of simulations maintain a steady initial flux profile before the long time-steps begin, whereas the latter show a symmetric radial oscillation across the first four time-points. This appears most likely due to these simulations resolving a physical xenon transient.

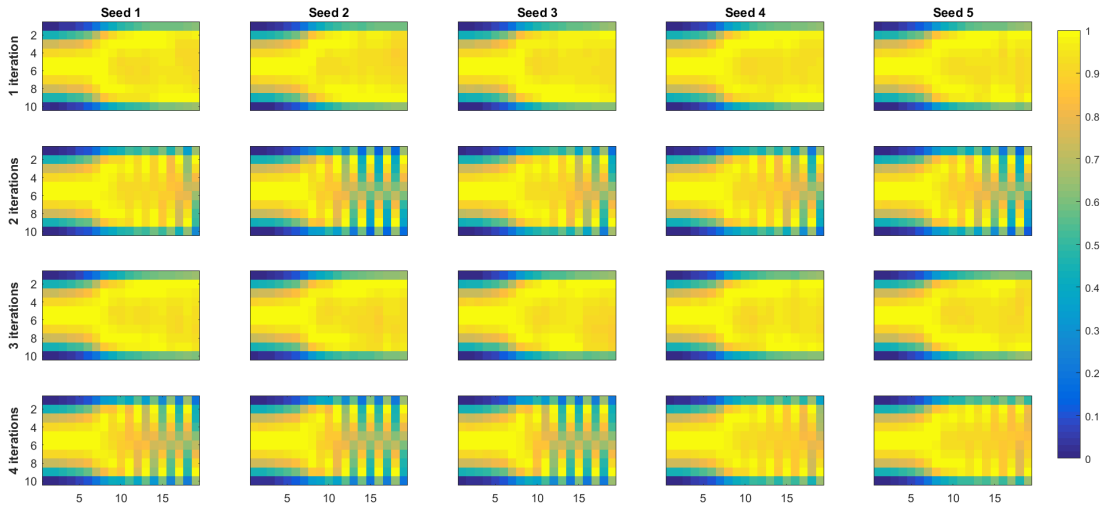


Fig. 3.13 Flux profile over time for a PWR fuel pin with axially vacuum boundaries using the LE/LI scheme with 10 substeps, enforcing xenon equilibrium, and varying the number of corrector iterations in various instances. The horizontal axes on each graph are the burn-up time-points given in Table 3.2, while the vertical axes are the flux values at a given axial node from 1 to 10, with the fluxes at a given time-point normalised by the maximum value.

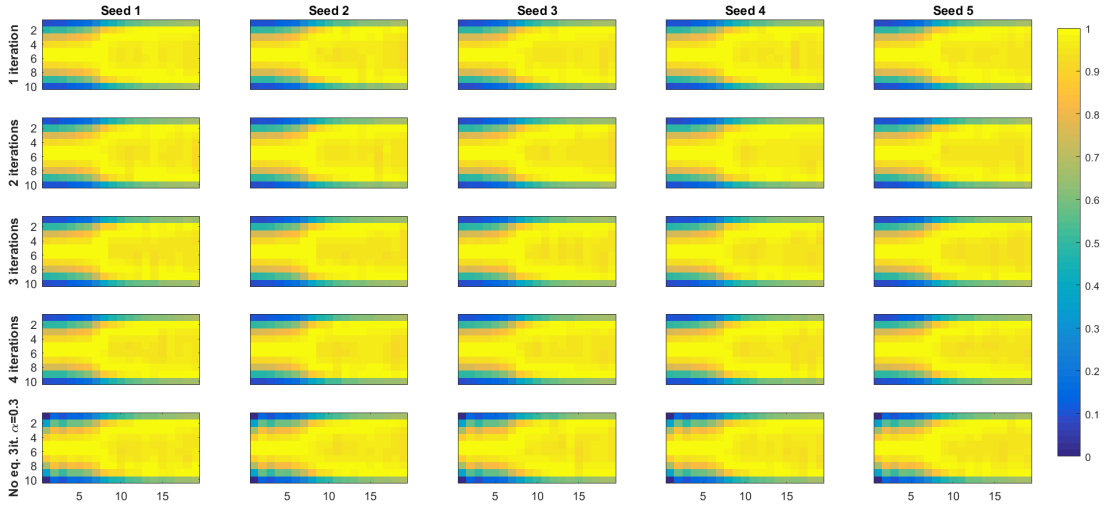


Fig. 3.14 Flux profile over time for a PWR fuel pin with axially vacuum boundaries using the LE/LI scheme with 10 substeps, enforcing xenon equilibrium, applying a relaxation of $\alpha = 0.5$, and varying the number of corrector iterations in various instances. The bottom row shows the same calculations performed without enforcing xenon equilibrium, using $\alpha = 0.3$, and performing 3 corrector iterations. The horizontal axes on each graph are the burn-up time-points given in Table 3.2, while the vertical axes are the flux values at a given axial node from 1 to 10, with the fluxes at a given time-point normalised by the maximum value.

3.5 Summary

This chapter has investigated various PC schemes where the corrector step is iterated and relaxed. Even when using Monte Carlo settings to reduce correlation, non-physical behaviour still develops with burn-up for moderate time-step lengths. For a variety of PC schemes, iterating on the corrector step does not improve stability, even appearing to develop non-physical behaviour more quickly. Relaxation on the corrector step was proposed, given the success of the stochastic approximation. It was noted that the stochastic approximation was motivated by the assumption that noise in the Monte Carlo solver substantially interferes with achieving a stable solution and does not necessarily provide an optimal relaxation schedule. Hence, the use of several fixed relaxation factors was investigated, applied to each PC scheme, and for multiple corrector iterations. This proved successful in achieving stable solutions, though more aggressive relaxation was required for higher-order schemes. Furthermore, it was demonstrated that one can use reduced Monte Carlo settings on the iterated corrector steps for computational efficiency while maintaining stability, as has been done with SI schemes. Finally, even when subject to substantial correlation, the fixed relaxation factor appears to achieve a more stable solution. Overall – provided a reasonable choice of relaxation factor is made – using a fixed relaxation factor appears more computationally efficient than applying the stochastic approximation. Finally, the effect of iterating on the corrector step was investigated while enforcing xenon equilibrium. Intriguingly, numerical instability resulted, but only when performing an even number of corrector iterations; if there was a reason to iterate corrector step depletion (such as when converging additional physics), such instabilities might manifest in a practical application. However, applying a relaxation to the corrector step prevents this behaviour.

Chapter 4

Von Neumann stability analysis of predictor-corrector depletion

Stability analysis allows the characteristics of numerical schemes to be interrogated, providing better understanding of their behaviour and informing their usage. In extremely successful cases, they may allow the development of heuristics for when to apply implicit schemes, or which time-step to choose, or may even allow automation of time-step selection. In the neutronics/depletion realm, the work of Densmore et al. [24] is where stability analysis begins and ends. However, their analysis is limited in a number of ways: it is restricted to a 1D, mono-energetic, diffusion problem, and is depleted using the explicit Euler scheme. This chapter intends to relax the last of these simplifications and extend their stability analysis to more sophisticated methods of coupling neutronics and depletion. In beginning this analysis, the chapter borrows heavily from their preceding work in order to obtain new results. Although the analysis is applied to a relatively simple system, it is hoped that future work will extend it to be useful in more sophisticated cases.

4.1 Governing equations

The neutron diffusion equation is a simplification of the transport equation, obtained by integrating the transport equation over angles and applying Fick's Law – it is usually covered in a first course in reactor physics [25]. A 1D mono-energetic diffusion equation in eigenvalue form, subject to isotropic scattering, can be written as:

$$-\frac{1}{3\Sigma_{\mathrm{t}}}\frac{\partial^2\phi}{\partial x^2}+\Sigma_{\mathrm{a}}\phi=\frac{1}{k}\nu\Sigma_{\mathrm{f}}\phi \tag{4.1}$$

Each of these terms is familiar: Σ_t is the total macroscopic cross-section, ϕ is the scalar flux, x is the spatial variable, Σ_a the absorption cross-section, k the eigenvalue, ν the average neutron production from fission, and Σ_f is the macroscopic fission cross-section. The domain extends from 0 to L, and the diffusion equation is subject to reflective boundary conditions:

$$\left.\frac{\partial \phi}{\partial x}\right|_{x=0} = \left.\frac{\partial \phi}{\partial x}\right|_{x=L} = 0 \tag{4.2}$$

The Bateman equation is given by:

$$\frac{\partial N}{\partial t} = (\mathbf{\Sigma}\phi + \mathbf{\Lambda})N \tag{4.3}$$

N is the time-dependent vector field of nuclide densities, t is the time variable, $\mathbf{\Sigma}$ is the matrix of transmutation and fission microscopic cross-sections, while $\mathbf{\Lambda}$ is the matrix accounting for nuclear decay. This is subject to the initial condition at time $t = 0$:

$$N(x,0) = N_0 \tag{4.4}$$

The diffusion solution is evaluated at a number of discrete time-points, denoted by the index n, such that the diffusion solution at a given time-point is obtained by solving:

$$-\frac{1}{3\Sigma_{\mathrm{t},n}}\frac{\partial^2 \phi_n}{\partial x^2} + \Sigma_{\mathrm{a},n}\phi_n = \frac{1}{k_n}\nu\Sigma_{\mathrm{f},n}\phi_n \tag{4.5}$$

Densmore et al. discretise the Bateman equation by an explicit Euler scheme in their paper:

$$N_{n+1} = [\mathbf{I} + \Delta t(\mathbf{\Sigma}\phi_n + \mathbf{\Lambda})]N_n \tag{4.6}$$

In the above equation, $\mathbf{I}$ is the identity matrix, and Δt is the time-step length – this is the Bateman matrix exponential truncated to first order. This chapter extends the previous analysis by treating a predictor-corrector time discretisation instead. Specifically, the above equation now only provides the intermediate $n + \frac{1}{2}$ nuclide density vector. Rewriting it to have this new intermediate index it becomes:

$$N_{n+\frac{1}{2}} = [\mathbf{I} + \Delta t(\mathbf{\Sigma}\phi_n + \mathbf{\Lambda})]N_n \tag{4.7}$$

This corresponds to the EOS nuclide density often referred to when discussing predictor-corrector depletion. The corresponding intermediate flux solution is obtained from the same

equation as before:

$$-\frac{1}{3\Sigma_{\mathrm{t},n+\frac{1}{2}}}\frac{\partial^2\phi_{n+\frac{1}{2}}}{\partial x^2}+\Sigma_{\mathrm{a},n+\frac{1}{2}}\phi_{n+\frac{1}{2}}=\frac{1}{k_{n+\frac{1}{2}}}\nu\Sigma_{\mathrm{f},n+\frac{1}{2}}\phi_{n+\frac{1}{2}} \tag{4.8}$$

The corrector depletion equation is:

$$N_{n+1}=\left[\mathbf{I}+\Delta t\left(\mathbf{\Sigma}\frac{\phi_n+\phi_{n+\frac{1}{2}}}{2}+\mathbf{\Lambda}\right)\right]N_n \tag{4.9}$$

Were the problem not mono-energetic, the flux-averaging during the corrector step would not be adequate – if the one-group cross-sections had a dependence upon the neutron spectrum (as in reality), then, instead, the reaction rates would be averaged during the corrector step. Prior to decomposing the solutions into Fourier modes, one should note that the flux should be spatially uniform and normalised to power at all times:

$$\phi_n=\frac{P}{\Sigma_{\mathrm{f},n}L} \tag{4.10}$$

Furthermore, the eigenvalue should be given simply by:

$$k_n=\frac{\nu\Sigma_{\mathrm{f},n}}{\Sigma_{\mathrm{a},n}} \tag{4.11}$$

4.2 Stability analysis

Following the Von Neumann stability analysis applied by Densmore et al., the variables are rewritten in terms of small perturbations:

$$N_n=N_0+\delta N_n \tag{4.12}$$

$$\phi_n=\phi_0+\delta\phi_n \tag{4.13}$$

$$k_n=k_0+\delta k \tag{4.14}$$

These expressions are inserted into the discretised equations above, expanded, and linearised in the perturbation terms. Doing this explicitly, the diffusion equation can be written as:

$$\begin{aligned} &-\frac{1}{3(\Sigma_{\mathrm{t},0}+\delta\Sigma_{\mathrm{t},n})}\frac{\partial^2(\phi_0+\delta\phi_n)}{\partial x^2} \\ &+\left[\Sigma_{\mathrm{a},0}+\delta\Sigma_{\mathrm{a},n}-\frac{1}{k_0+\delta k_n}\left(\nu\Sigma_{\mathrm{f},0}+\nu\delta\Sigma_{\mathrm{f},n}\right)\right](\phi_0+\delta\phi_n)=0 \end{aligned} \tag{4.15}$$

Expanding the denominators and neglecting second order terms gives:

$$\begin{aligned} -\frac{1}{3\Sigma_{\mathrm{t},0}}\frac{\partial^2\phi_0}{\partial x^2}-\frac{1}{3\Sigma_{\mathrm{t},0}}\frac{\partial^2\delta\phi_n}{\partial x^2}+\frac{\delta\Sigma_{\mathrm{t}}}{3\Sigma_{\mathrm{t},0}^2}\frac{\partial^2\phi_0}{\partial x^2}+\Sigma_{\mathrm{a},0}(\phi_0+\delta\phi_n)+\delta\Sigma_a\phi_0 \\ +\frac{\delta k_n}{k_0^2}\nu\Sigma_{\mathrm{f},0}\phi_0-\frac{1}{k_0}\left(\nu\Sigma_{\mathrm{f},0}\phi_0+\nu\delta\Sigma_{\mathrm{f},n}\phi_0+\nu\Sigma_{\mathrm{f},0}\delta\phi_n\right)=0 \end{aligned} \tag{4.16}$$

Noting that the ϕ_0 is spatially constant and the definition of k_0 as the ratio of the initial neutron production and absorption cross-sections, the above simplifies to:

$$-\frac{1}{3\Sigma_{\mathrm{t},0}}\frac{\partial^2\delta\phi_n}{\partial x^2}=\left(\frac{\delta\Sigma_{\mathrm{f},n}}{\Sigma_{\mathrm{f},0}}-\frac{\delta\Sigma_{\mathrm{a},n}}{\Sigma_{\mathrm{a},0}}-\frac{\delta k_n}{k_0}\right)\Sigma_{\mathrm{a},0}\phi_0 \tag{4.17}$$

Repeating the same for the nuclide density field:

$$N_0+\delta N_{n+\frac{1}{2}}=\left[\mathbf{I}+\Delta t(\mathbf{\Sigma}(\phi_0+\delta\phi_n)+\mathbf{\Lambda})\right](N_0+\delta N_n) \tag{4.18}$$

The N_0 term on the left-hand side cancels with the $\mathbf{I}N_0$ term on the right-hand side, leaving:

$$\delta N_{n+\frac{1}{2}}=\left[\mathbf{I}+\Delta t(\mathbf{\Sigma}\phi_0+\mathbf{\Lambda})\right]\delta N_n+\Delta t(\mathbf{\Sigma}\phi_0+\mathbf{\Lambda})N_0+\Delta t\mathbf{\Sigma}\delta\phi_n N_0 \tag{4.19}$$

The intermediate flux error, $\delta\phi_{n+\frac{1}{2}}$, is obtained from an identical equation as Eq. (4.17), but using the $n+\frac{1}{2}$ nuclide density field. Finally, the new nuclide density vector is:

$$\begin{aligned} \delta N_{n+1}&=\left[\mathbf{I}+\Delta t\left(\mathbf{\Sigma}\phi_0+\mathbf{\Lambda}\right)\right]\delta N_n \\ &+\Delta t(\mathbf{\Sigma}\phi_0+\mathbf{\Lambda})N_0+\frac{\Delta t}{2}\mathbf{\Sigma}\delta\phi_n N_0+\frac{\Delta t}{2}\mathbf{\Sigma}\delta\phi_{n+\frac{1}{2}}N_0 \end{aligned} \tag{4.20}$$

From here, the classic Von Neumann stability analysis decomposes these perturbations in the fields into Fourier modes which form a complete set on the interval in question:

$$\delta\phi_n(x)=\sum_{m=0}^{\infty}\delta\phi_{n,m}\cos\left(\frac{m\pi x}{L}\right) \tag{4.21}$$

$$\delta N_n(x) = \sum_{m=0}^{\infty} \delta N_{n,m} \cos\left(\frac{m\pi x}{L}\right) \tag{4.22}$$

The orthogonality of each Fourier mode can be used to analyse their evolution separately. The diffusion equation for the m-th mode becomes:

$$\frac{1}{3\Sigma_{\mathrm{t},0}} \delta\phi_{n,m} \left(\frac{m\pi}{L}\right)^2 = \left(\frac{\sigma_{\mathrm{f}}^T}{\Sigma_{\mathrm{f},0}} \delta N_{n,m} - \frac{\sigma_{\mathrm{a}}^T}{\Sigma_{\mathrm{a},0}} \delta N_{n,m} - \frac{\delta k_n}{k_0} \delta_{m,0}\right) \Sigma_{\mathrm{a},0}\phi_0 \tag{4.23}$$

As Densmore et al. highlight, the $\delta_{m,0}$ term exists as the spatially constant term only, contributing to the zeroth solution mode. When $m > 0$, Eq. (4.23) can be solved for the flux mode resulting from a nuclide density vector mode:

$$\delta\phi_{n,m} = \phi_0 \left(\frac{L}{m\pi L_D}\right)^2 \left(\frac{\sigma_{\mathrm{f}}^T}{\Sigma_{\mathrm{f},0}} - \frac{\sigma_{\mathrm{a}}^T}{\Sigma_{\mathrm{a},0}}\right) \delta N_{n,m} \tag{4.24}$$

The L_D term is the diffusion length, equal to $\frac{1}{\sqrt{3\Sigma_{\mathrm{t},0}\Sigma_{\mathrm{a},0}}}$. The first nuclide density error vector can be expressed as:

$$\begin{aligned} \delta N_{n+\frac{1}{2},m} &= \left[\mathbf{I} + \Delta t(\mathbf{\Sigma}\phi_0 + \mathbf{\Lambda})\right] \delta N_{n,m} \\ &+ \Delta t(\mathbf{\Sigma}\phi_0 + \mathbf{\Lambda}) N_0 \delta_{m,0} + \Delta t \mathbf{\Sigma} \delta\phi_{n,m} N_0 \end{aligned} \tag{4.25}$$

As with the decomposition of the diffusion equation, this, too, possesses a delta term which contributes only to the $m = 0$ mode. Inserting the expression for the flux modes from Eq. (4.24), for $m > 0$ this becomes:

$$\begin{aligned} \delta N_{n+\frac{1}{2},m} &= \left[\mathbf{I} + \Delta t(\mathbf{\Sigma}\phi_0 + \mathbf{\Lambda})\right] \delta N_{n,m} \\ &+ \Delta t \mathbf{\Sigma} N_0 \phi_0 \left(\frac{L}{m\pi L_D}\right)^2 \left(\frac{\sigma_{\mathrm{f}}^T}{\Sigma_{\mathrm{f},0}} - \frac{\sigma_{\mathrm{a}}^T}{\Sigma_{\mathrm{a},0}}\right) \delta N_{n,m} \end{aligned} \tag{4.26}$$

Or, more succinctly:

$$\delta N_{n+\frac{1}{2},m} = \mathbf{M}_m \delta N_{n,m} \tag{4.27}$$

With:

$$\mathbf{M}_m = \left[\left[\mathbf{I} + \Delta t(\mathbf{\Sigma}\phi_0 + \mathbf{\Lambda})\right] + \Delta t \mathbf{\Sigma} N_0 \phi_0 \left(\frac{L}{m\pi L_D}\right)^2 \left(\frac{\sigma_{\mathrm{f}}^T}{\Sigma_{\mathrm{f},0}} - \frac{\sigma_{\mathrm{a}}^T}{\Sigma_{\mathrm{a},0}}\right)\right] \tag{4.28}$$

For the $m = 1$ mode, the dependence upon the dominance ratio, ρ, of the problem can be made explicit. Recall ρ is defined as the ratio of the first eigenvalue to the fundamental eigenvalue, $\frac{k_1}{k_0}$. For this simple problem, the initial dominance ratio, ρ_0, can be obtained

analytically as:

$$\rho_0 = \frac{1}{1+\left(\frac{\pi L_D}{L}\right)^2} \tag{4.29}$$

Therefore, $\mathbf{M}_1$ can be expressed as:

$$\mathbf{M}_1 = \left[[\mathbf{I}+\Delta t(\mathbf{\Sigma}\phi_0+\mathbf{\Lambda})]+\Delta t\mathbf{\Sigma}N_0\phi_0\frac{\rho_0}{1-\rho_0}\left(\frac{\sigma_{\mathrm{f}}^T}{\Sigma_{\mathrm{f},0}}-\frac{\sigma_{\mathrm{a}}^T}{\Sigma_{\mathrm{a},0}}\right)\right] \tag{4.30}$$

In order to conclude the analysis from Densmore et al. before extending it further, obtaining the solutions for the $m=0$ modes should also be demonstrated. As stated above, the $m=0$ flux mode cannot be obtained directly from Eq. (4.24). Instead, the power normalisation is used, and, in particular, Eq. (4.10) is expanded with the $m=0$ flux and nuclide density vector modes. This gives:

$$(\phi_0+\delta\phi_{n,0})\,\sigma_{\mathrm{f}}^T\,(N_0+\delta N_{n,0}) = P/L \tag{4.31}$$

The $\phi_0\Sigma_{\mathrm{f},0}$ term is equal to P/L. Therefore, neglecting the second-order terms:

$$\delta\phi_{n,0}\Sigma_{\mathrm{f},0}^T+\sigma_{\mathrm{f}}^T\,\delta N_{n,0}\phi_0 = 0 \tag{4.32}$$

Rearranging gives:

$$\delta\phi_{n,0} = -\frac{\phi_0\sigma_{\mathrm{f}}^T}{\Sigma_{\mathrm{f},0}}\delta N_{n,0} \tag{4.33}$$

This can be inserted into Eq. (4.25) to give:

$$\begin{aligned}\delta N_{n+\frac{1}{2},0} &= [\mathbf{I}+\Delta t(\mathbf{\Sigma}\phi_0+\mathbf{\Lambda})]\,\delta N_{n,0}\\ &+\Delta t(\mathbf{\Sigma}\phi_0+\mathbf{\Lambda})N_0-\Delta t\mathbf{\Sigma}N_0\frac{\phi_0\sigma_{\mathrm{f}}^T}{\Sigma_{\mathrm{f},0}}\delta N_{n,0}\end{aligned} \tag{4.34}$$

Or:

$$\delta N_{n+\frac{1}{2},0} = \mathbf{M}_0\delta N_{n,0}+\Delta t(\mathbf{\Sigma}\phi_0+\mathbf{\Lambda})N_0 \tag{4.35}$$

With:

$$\mathbf{M}_0 = \left[[\mathbf{I}+\Delta t(\mathbf{\Sigma}\phi_0+\mathbf{\Lambda})]-\Delta t\mathbf{\Sigma}N_0\frac{\phi_0\sigma_{\mathrm{f}}^T}{\Sigma_{\mathrm{f},0}}\right] \tag{4.36}$$

These results are identical to those obtained by Densmore et al. if the $N_{n+\frac{1}{2}}$ value is accepted as the updated nuclide density vector for the next time-step. Extending the analysis to the PC scheme requires also considering the corrector update equation for the nuclide density vector.

These corrector nuclide density vector modes are given by:

$$\begin{aligned}\delta N_{n+1,m} &= [\mathbf{I} + \Delta t\,(\mathbf{\Sigma}\phi_0 + \mathbf{\Lambda})]\delta N_{n,m} \\ &+ \Delta t(\mathbf{\Sigma}\phi_0 + \mathbf{\Lambda})N_0\delta_{m,0} + \frac{\Delta t}{2}\mathbf{\Sigma}N_0\delta\phi_{n,m} + \frac{\Delta t}{2}\mathbf{\Sigma}N_0\delta\phi_{n+\frac{1}{2},m}\end{aligned} \tag{4.37}$$

Considering the $m > 0$ modes first gives:

$$\begin{aligned}\delta N_{n+1,m} &= [\mathbf{I} + \Delta t\,(\mathbf{\Sigma}\phi_0 + \mathbf{\Lambda})]\delta N_{n,m} \\ &+ \frac{\Delta t}{2}\mathbf{\Sigma}N_0\left(\delta\phi_{n,m} + \delta\phi_{n+\frac{1}{2},m}\right)\end{aligned} \tag{4.38}$$

The n-th flux modes are obtained from Eq. (4.24). The $(n+\frac{1}{2})$-th modes can be obtained straightforwardly given Eq. (4.24) and (4.26):

$$\delta\phi_{n+\frac{1}{2},m} = \mathbf{\Phi}_m\left[\mathbf{B} + \Delta t\mathbf{\Sigma}N_0\mathbf{\Phi}_m\right]\delta N_{n,m} \tag{4.39}$$

Where, for brevity, two new operators are defined:

$$\mathbf{\Phi}_m = \phi_0\left(\frac{L}{m\pi L_D}\right)^2\left(\frac{\sigma_{\mathrm{f}}^T}{\Sigma_{\mathrm{f},0}} - \frac{\sigma_{\mathrm{a}}^T}{\Sigma_{\mathrm{a},0}}\right) \tag{4.40}$$

And

$$\mathbf{B} = [\mathbf{I} + \Delta t(\mathbf{\Sigma}\phi_0 + \mathbf{\Lambda})] \tag{4.41}$$

Hence, Eq. (4.38) becomes

$$\delta N_{n+1,m} = \mathbf{M}_m\delta N_{n,m} \tag{4.42}$$

With:

$$\mathbf{M}_m = \left[\mathbf{B} + \frac{\Delta t}{2}\mathbf{\Sigma}N_0\mathbf{\Phi}_m\left[\mathbf{I} + \mathbf{B} + \Delta t\mathbf{\Sigma}N_0\mathbf{\Phi}_m\right]\right] \tag{4.43}$$

The corresponding equation for the explicit Euler scheme, using the same notation, is:

$$\delta N_{n+1,m} = \left[\mathbf{B} + \Delta t\mathbf{\Sigma}N_0\mathbf{\Phi}_m\right]\delta N_{n,m} \tag{4.44}$$

It can be seen that, in the limit of a short time-step where Δt^2 terms can be neglected, the matrix describing the evolution of the PC Fourier modes approaches that of the Euler scheme.

To the $m = 0$ mode, this begins with:

$$\begin{aligned}\delta N_{n+1,0} &= [\mathbf{I} + \Delta t\,(\mathbf{\Sigma}\phi_0 + \mathbf{\Lambda})]\delta N_{n,0} \\ &+ \Delta t(\mathbf{\Sigma}\phi_0 + \mathbf{\Lambda})N_0 + \frac{\Delta t}{2}\mathbf{\Sigma}N_0\delta\phi_{n,0} + \frac{\Delta t}{2}\mathbf{\Sigma}N_0\delta\phi_{n+\frac{1}{2},0}\end{aligned} \tag{4.45}$$

The expression for the flux mode at $n+\frac{1}{2}$ combines Eqs. (4.33) and (4.34) to give:

$$\delta\phi_{n+\frac{1}{2},0} = \mathbf{\Phi}_0\left[\mathbf{B} + \Delta t\mathbf{\Sigma}N_0\mathbf{\Phi}_0\right]\delta N_{n,0} + \mathbf{\Phi}_0\left[\mathbf{B} - \mathbf{I}\right]N_0 \tag{4.46}$$

Again for brevity:

$$\mathbf{\Phi}_0 = -\frac{\phi_0\sigma_{\mathrm{f}}^T}{\Sigma_{\mathrm{f},0}} \tag{4.47}$$

With this, Eq. (4.45) can be written:

$$\delta N_{n+1,0} = \mathbf{M}_0\delta N_{n,0} + \mathbf{K}_0 N_0 \tag{4.48}$$

With:

$$\mathbf{M}_0 = \left[\mathbf{B} + \frac{\Delta t}{2}\mathbf{\Sigma}N_0\mathbf{\Phi}_0\left[\mathbf{I} + \mathbf{B} + \Delta t\mathbf{\Sigma}N_0\mathbf{\Phi}_0\right]\right] \tag{4.49}$$

And:

$$\mathbf{K}_0 = \left[\mathbf{I} + \frac{\Delta t}{2}\mathbf{\Sigma}N_0\mathbf{\Phi}_0\right]\left[\mathbf{B} - \mathbf{I}\right] \tag{4.50}$$

On the other hand, the explicit Euler expression is:

$$\delta N_{n+1,0} = \left[\mathbf{B} + \Delta t\mathbf{\Sigma}N_0\mathbf{\Phi}_0\right]\delta N_{n,0} + \left[\mathbf{B} - \mathbf{I}\right]N_0 \tag{4.51}$$

4.2.1 Multiple corrector iterations

This analysis can be extended further to performing more than one corrector iteration. Consider adding a second corrector step – the previous $n+1$ values are now denoted with a superscript, rather than using the $n+\frac{1}{2}$ notation. The $(n+1)$-th nuclide density is obtained from:

$$N_{n+1}^{(2)} = \left[\mathbf{I} + \Delta t\left(\mathbf{\Sigma}\frac{\phi_n + \phi_{n+1}^{(1)}}{2} + \mathbf{\Lambda}\right)\right]N_n \tag{4.52}$$

The flux solution after the first corrector step, $\phi_{n+1}^{(1)}$, is obtained from the corresponding nuclide density vector as before, with the same operator relating the nuclide density modes to the flux modes, $\mathbf{\Phi}_m$ or $\mathbf{\Phi}_0$. Furthermore, the expression for $\delta N_{n+1,m}^{(1)}$ is simply one of

Eqs. (4.42) or (4.48). For the former ($m > 0$) case, one obtains:

$$\delta N^{(2)}_{n+1,m} = \left[\mathbf{B} + \frac{\Delta t}{2}\mathbf{\Sigma} N_0 \mathbf{\Phi}_m \left[\mathbf{I} + \mathbf{M}^{(1)}_m\right]\right] \delta N_{n,m} \tag{4.53}$$

Where $\mathbf{M}^{(1)}_m$ is given by Eq. (4.43). This can be used to initialise a recurrence relation for J corrector iterations, giving:

$$\delta N^{(J)}_{n+1,m} = \mathbf{M}^{(J)}_m \delta N_{n,m} \tag{4.54}$$

And:

$$\mathbf{M}^{(J)}_m = \left[\mathbf{B} + \frac{\Delta t}{2}\mathbf{\Sigma} N_0 \mathbf{\Phi}_m \left[\mathbf{I} + \mathbf{M}^{(J-1)}_m\right]\right] \tag{4.55}$$

The same logic applies to obtaining the expression for the $m = 0$ mode subject to multiple corrector iterations. Making use of the expressions in Eqs. (4.49) and (4.50), one obtains:

$$\begin{aligned}\delta N^{(2)}_{n+1,0} = &\left[\mathbf{B} + \frac{\Delta t}{2}\mathbf{\Sigma} N_0 \mathbf{\Phi}_0 \left[\mathbf{I} + \mathbf{M}_0\right]\right] \delta N_{n,0} \\ &+ \left[\mathbf{B} - \mathbf{I} + \frac{\Delta t}{2}\mathbf{\Sigma} N_0 \mathbf{\Phi}_0 \mathbf{K}_0\right] N_0\end{aligned} \tag{4.56}$$

Once again, for J corrector iterations, this is convenient to write as a recurrence relation, initialised by Eqs. (4.49) and (4.50), such that one has:

$$\delta N^{(J)}_{n+1,0} = \mathbf{M}^{(J)}_0 \delta N_{n,0} + \mathbf{K}^{(J)}_0 N_0 \tag{4.57}$$

Where:

$$\mathbf{M}^{(J)}_0 = \left[\mathbf{B} + \frac{\Delta t}{2}\mathbf{\Sigma} N_0 \mathbf{\Phi}_0 \left[\mathbf{I} + \mathbf{M}^{(J-1)}_0\right]\right] \tag{4.58}$$

And:

$$\mathbf{K}^{(J)}_0 = \left[\mathbf{B} - \mathbf{I} + \frac{\Delta t}{2}\mathbf{\Sigma} N_0 \mathbf{\Phi}_0 \mathbf{K}^{(J-1)}_0\right] \tag{4.59}$$

4.2.2 Relaxation on the corrector iteration

Finally, one might ask how relaxation affects the stability of the corrector iteration. In particular, the corrector equation is modified such that performing J corrector iterations is

expressed as:

$$N_{n+1}^{(J)} = \alpha \left[\mathbf{I} + \Delta t \left(\mathbf{\Sigma} \frac{\phi_n + \phi_{n+1}^{(J-1)}}{2} + \mathbf{\Lambda} \right) \right] N_n + (1-\alpha) N_{n+1}^{(J-1)} \tag{4.60}$$

As in Chapter 3, α is the relaxation factor, taking a value between 0 and 1. Beginning, as in the previous section, with a single corrector iteration, one has:

$$N_{n+1}^{(1)} = \alpha \left[\mathbf{I} + \Delta t \left(\mathbf{\Sigma} \frac{\phi_n + \phi_{n+1}^{(0)}}{2} + \mathbf{\Lambda} \right) \right] N_n + (1-\alpha) N_{n+1}^{(0)} \tag{4.61}$$

Following the same steps as before, for $m > 0$ and a single corrector iteration, one ultimately obtains:

$$\delta N_{n+1,m}^{(1)} = \mathbf{M}_m \delta N_{n,m} \tag{4.62}$$

With

$$\begin{aligned} \mathbf{M}_m = & \alpha \left(\mathbf{B} + \frac{\Delta t}{2} \mathbf{\Sigma} N_0 \mathbf{\Phi}_m \left[\mathbf{I} + \mathbf{B} + \Delta t \mathbf{\Sigma} N_0 \mathbf{\Phi}_m \right] \right) \\ & + (1-\alpha) \left(\mathbf{B} + \Delta t \mathbf{\Sigma} N_0 \mathbf{\Phi}_m \right) \end{aligned} \tag{4.63}$$

The most convenient way to write this for multiple corrector iterations is to use the matrix $\mathbf{M}_m$ in a recurrence relation, such that, for $J > 1$ corrector iterations, one has:

$$\delta N_{n+1,m}^{(J)} = \mathbf{M}_m^{(J)} \delta N_{n,m} \tag{4.64}$$

Where

$$\mathbf{M}_m^{(J)} = \left[\alpha \left(\mathbf{B} + \frac{\Delta t}{2} \mathbf{\Sigma} N_0 \mathbf{\Phi}_m \right) + \left((1-\alpha) \mathbf{I} + \alpha \frac{\Delta t}{2} \mathbf{\Sigma} N_0 \mathbf{\Phi}_m \right) \mathbf{M}_m^{(J-1)} \right] \tag{4.65}$$

With $\mathbf{M}_m^{(1)}$ given by Eq. (4.63). From expanding Eq. (4.61), the $m = 0$ mode is given by:

$$\begin{aligned} \delta N_{n+1,0}^{(1)} = & \alpha \left[\mathbf{B} + \frac{\Delta t}{2} \mathbf{\Sigma} N_0 \mathbf{\Phi}_0 \right] \delta N_{n,0} \\ & + \left[(1-\alpha) \mathbf{I} + \alpha \frac{\Delta t}{2} \mathbf{\Sigma} N_0 \mathbf{\Phi}_0 \right] \delta N_{n+1,0}^{(0)} \\ & + \left[\mathbf{B} - \mathbf{I} \right] N_0 \end{aligned} \tag{4.66}$$

The expression for $\delta N^{(0)}_{n+1,0}$ is the same as before from Eq. (4.51), giving an expression for the first corrector iteration as:

$$\delta N^{(1)}_{n+1,0} = \mathbf{M}_0 \delta N_{n,0} + \mathbf{K}_0 N_0 \tag{4.67}$$

Where

$$\mathbf{M}_0 = \left[\mathbf{B} + (1-\alpha)\Delta t \mathbf{\Sigma} N_0 \mathbf{\Phi}_0 + \alpha \frac{\Delta t}{2} \mathbf{\Sigma} N_0 \mathbf{\Phi}_0 \left[\mathbf{I} + \mathbf{B} + \Delta t \mathbf{\Sigma} N_0 \mathbf{\Phi}_0\right]\right] \tag{4.68}$$

$$\mathbf{K}_0 = \left[(2-\alpha)\mathbf{I} + \alpha \frac{\Delta t}{2} \mathbf{\Sigma} N_0 \mathbf{\Phi}_0\right] [\mathbf{B} - \mathbf{I}] \tag{4.69}$$

Extending this to J iterations results in a similar expression as for the $m > 0$ modes, with:

$$\delta N^{(J)}_{n+1,0} = \mathbf{M}^{(J)}_0 \delta N_{n,0} + \mathbf{K}^{(J)}_0 N_0 \tag{4.70}$$

With the recurrence relations for each operator as:

$$\mathbf{M}^{(L)}_0 = \left[\alpha \left[\mathbf{B} + \frac{\Delta t}{2} \mathbf{\Sigma} N_0 \mathbf{\Phi}_0\right] + \left[(1-\alpha)\mathbf{I} + \alpha \frac{\Delta t}{2} \mathbf{\Sigma} N_0 \mathbf{\Phi}_0\right] \mathbf{M}^{(J-1)}_0\right] \tag{4.71}$$

And:

$$\mathbf{K}^{(J)}_0 = \left[\mathbf{B} - \mathbf{I} + \left[(1-\alpha)\mathbf{I} + \alpha \frac{\Delta t}{2} \mathbf{\Sigma} N_0 \mathbf{\Phi}_0\right] \mathbf{K}^{(J-1)}_0\right] \tag{4.72}$$

The $\mathbf{M}^{(1)}_0$ and $\mathbf{K}^{(1)}_0$ terms are given by Eqs. (4.68) and (4.69), respectively. For each of these expressions for the matrices governing the growth of the Fourier modes, the analysis can be applied to the stochastic approximation in particular by letting α vary with each iteration as $\frac{1}{j}$ where j is the iteration number.

4.3 Numerical investigations

4.3.1 Two-nuclide system

To compute initial numerical results and examine the predictions of these equations, the same system as described by Densmore et al. will be used. Namely, a 3 m long problem, featuring a two-nuclide depletion system, containing a fissile fuel and fission product, without decay,

described by the transmutation matrix $\mathbf{\Sigma}$:

$$\mathbf{\Sigma} = \begin{bmatrix} -3000 & 0 \\ 3000 & -150 \end{bmatrix} \tag{4.73}$$

The initial nuclide density vector is given (in atoms/b/cm) by:

$$N_0 = \begin{bmatrix} 0.00025 \\ 0 \end{bmatrix} \tag{4.74}$$

The microscopic cross-sections for the system are (in barns):

$$\sigma_\mathrm{t} = \begin{bmatrix} 4000 \\ 150 \end{bmatrix} \quad \sigma_\mathrm{a} = \begin{bmatrix} 3000 \\ 150 \end{bmatrix} \quad \sigma_\mathrm{f} = \begin{bmatrix} 3000 \\ 0 \end{bmatrix} \tag{4.75}$$

Densmore et al. emphasise that the cross-sections were chosen to provide the solutions with realistic time-step lengths for depletion problems. Finally, $\nu = 2.3$, the power is normalised to 10 kW/cm^2, and fission produces 200 MeV of energy. When solving the problem numerically, it is done using a spatial discretisation of 150 meshes and the eigenvalue problem is solved using MATLAB's `eig` function.

For each of the variations on the PC scheme, the point of interest is the eigenvalue with the largest magnitude of the governing Fourier mode evolution matrices. This eigenvalue is affected by the choice of time-step length, Δt. For the two nuclide system, each matrix, for each solution mode, will have two eigenvalues. In all cases observed, one of these eigenvalues remains approximately constant at a value of 1. This implies neither a growth nor decay in the solution mode. The other eigenvalue, however, may vary substantially – it may be positive and greater than 1 (implying growth), negative and less than -1 (implying growth and oscillation) or have an absolute value less than 1 (thus decaying). This is the eigenvalue which will be examined here. For the explicit Euler scheme, Densmore et al. found that this eigenvalue was only ever negative and grew linearly with time-step length for all modes, with the $m = 1$ mode the most dominant [24].

Results for the simplest version of the PC scheme are shown in Fig. 4.1. The PC matrix given by Eq. (4.43) is quadratic in the time-step, and, likewise, so are the eigenvalues of the PC scheme. Furthermore, as opposed to the explicit Euler scheme which was found to only produce oscillatory solutions, the PC scheme has only positive eigenvalues. This corresponds with the PC behaviour observed elsewhere in this thesis wherein flux tends to tilt to one side and remain there, rather than oscillating as the explicit Euler scheme was seen to do in Chapter 1. Furthermore, it appears that the PC scheme does not provide any additional

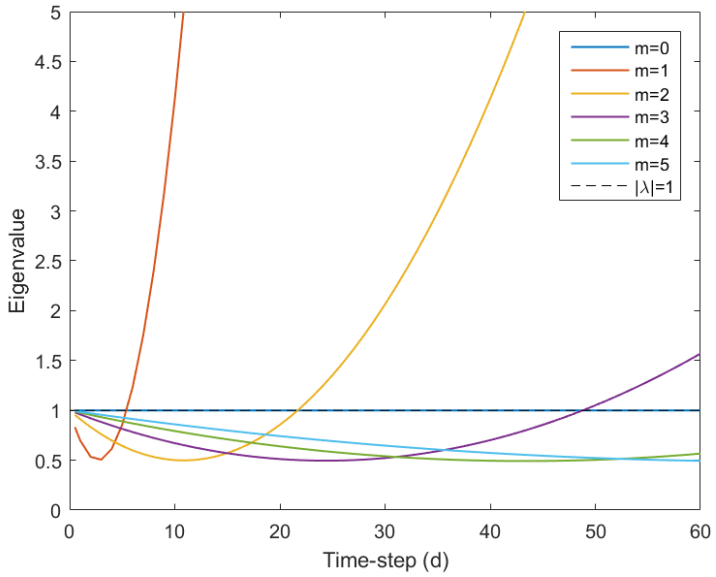
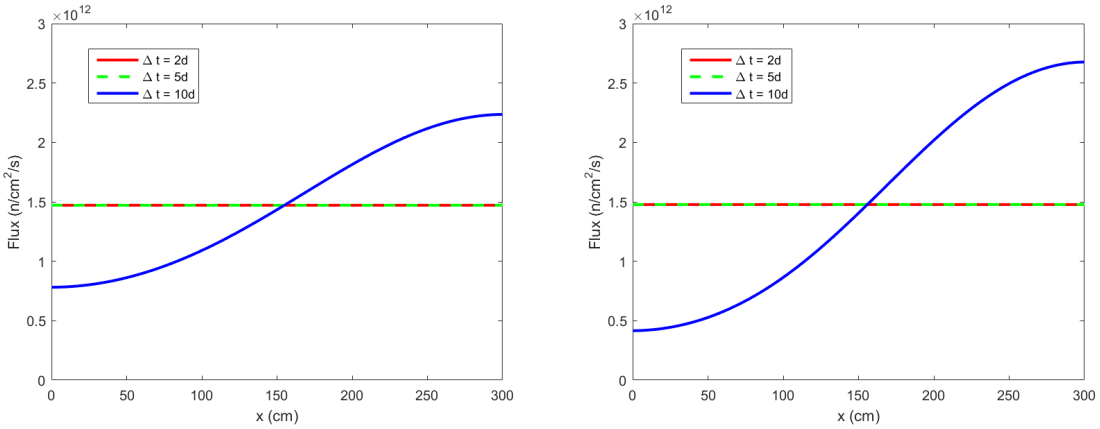


Fig. 4.1 Eigenvalues of the PC scheme with time-step.

stability as compared to the explicit Euler scheme according to this analysis. Examining the $m = 1$ mode, and solving Eqs. (4.28) and (4.43) for the minimum time-step length resulting in an eigenvalue, λ, such that $|\lambda| = 1$, both the explicit Euler and PC schemes have an identical unstable time-step of about 5.5 days. In fact, because of the quadratic dependence on time-step length, the magnitude of the PC eigenvalue grows faster than that of the explicit Euler scheme, resulting in unstable solution modes growing faster when the stable time-step has been exceeded.

In order to demonstrate that these predictions are borne out in simulation, the problem is explicitly solved using time-steps of 2, 5, and 10 days. The results for consecutive time-points (when taking 10 day time-steps) are shown in Fig. 4.2. These correspond well with the predictions of the stability analysis, i.e., the two shorter time-steps are stable, whereas the 10 day time-steps excite the first harmonic, amplifying it with time.

Next, iteration on the corrector step is considered, as in Chapter 3. Figs 4.3, 4.4, and 4.5 show the eigenvalues of the first few modes of the PC scheme using two, three, and four corrector iterations, respectively. The eigenvalues reproduce the behaviour previously seen, namely, they are negative for an odd number of iterations (giving oscillatory solutions) and positive for an even number of iterations (giving increasingly tilted solutions). Furthermore, there is no change in the stable time-step length when the number of corrector iterations is varied. In fact, due to the increasingly higher-degree dependence upon the time-step length with corrector iterations, performing additional iterations amplifies erroneous solution modes



(a) 160 days (b) 170 days

Fig. 4.2 PC flux solutions at consecutive time-points.

more dramatically for the same number of time-steps, as noted in Chapter 3. As with the explicit Euler and standard PC schemes, the first harmonic is the most unstable.

Finally, the stability of relaxing the corrector step is investigated. First, fixing the relaxation factor at 0.1 or 0.5 and performing various numbers of corrector iterations, the eigenvalues for the $m = 1$ mode are shown in Fig. 4.6. Relaxing the problem more aggressively increases stability, as does performing additional iterations. However, for a fixed relaxation factor, the stable time-step resulting from performing additional iterations approaches an asymptotic value which increases as the relaxation factor shrinks. One can also notice that performing an odd number of iterations can be more stable than performing slightly more iterations if the number is even. This can be seen by examining the stable time-step length in Fig. 4.6 for 7 and 10 iterations. This is only observed when there is a relaxation factor applied. The single iteration $m = 1$ mode eigenvalue with a relaxation factor of 0.1 has two stable regions in which $|\lambda| < 1$. This implies the possibility that other modes can be more unstable than the first. This was investigated by examining the other modes for a relaxation factor of 0.1 subject to a single iteration. The eigenvalues of the modes with time-step are shown in Fig. 4.7(a); this shows that with a time-step of about 50 days only the $m = 2$ mode is unstable. Furthermore, this mode is symmetric and so should be easily identifiable. Solving the problem explicitly using 50 day time-steps, one does find the symmetric instability to grow and oscillate with time, as shown in Fig. 4.7(b). This shows that the analysis was performed correctly and highlights that there are conceivable scenarios in which asymmetry is not definitive in identifying burn-up instabilities.

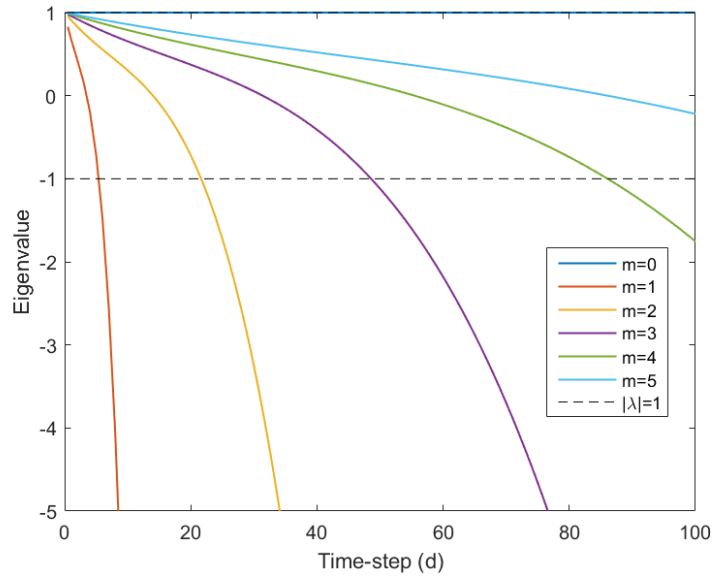


Fig. 4.3 Eigenvalues of the PC scheme with time-step when using two corrector iterations.

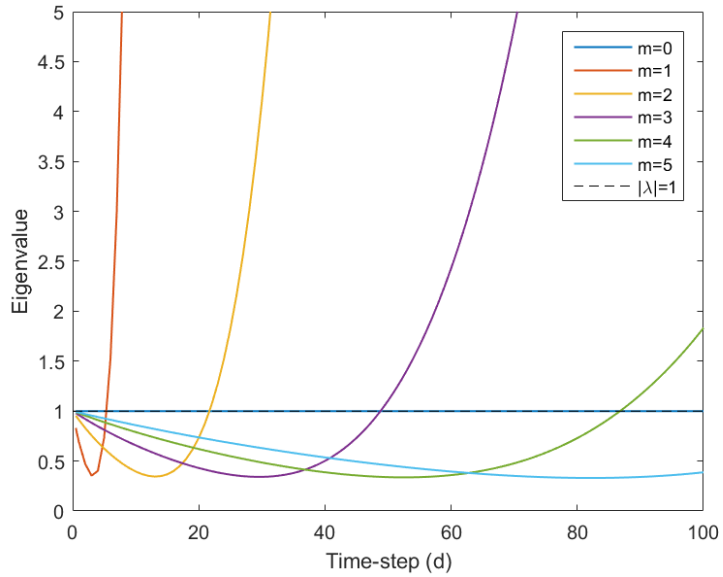


Fig. 4.4 Eigenvalues of the PC scheme with time-step when using three corrector iterations.

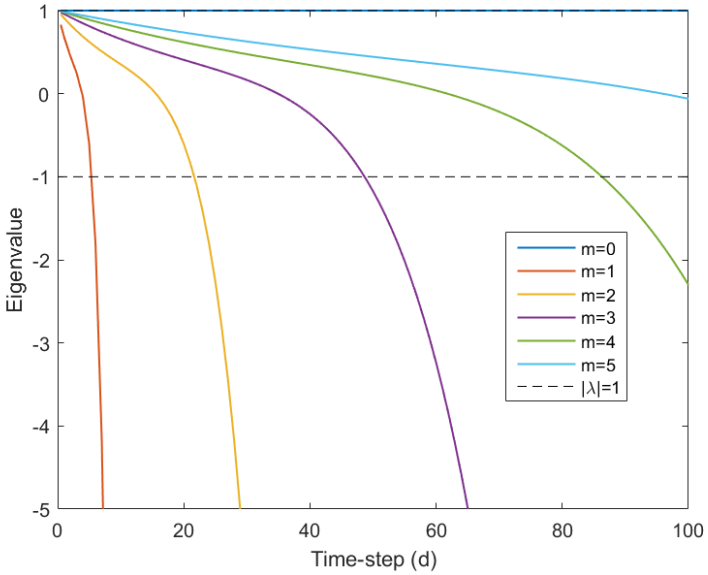
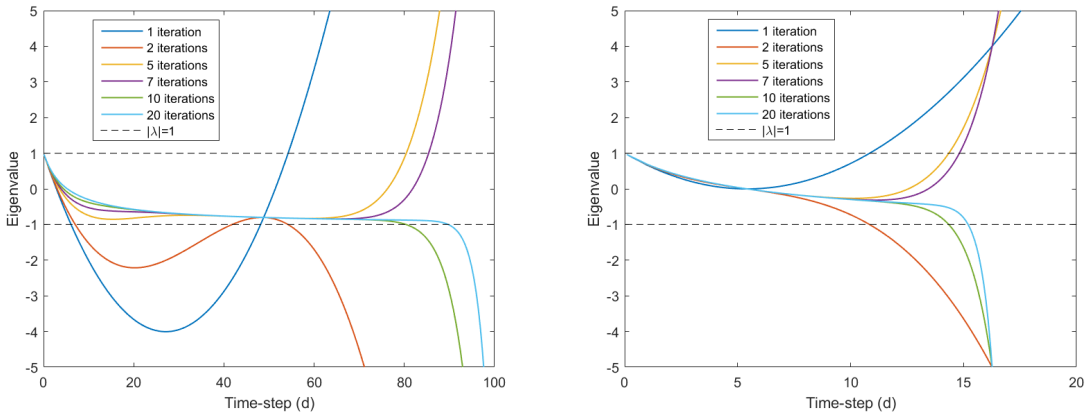
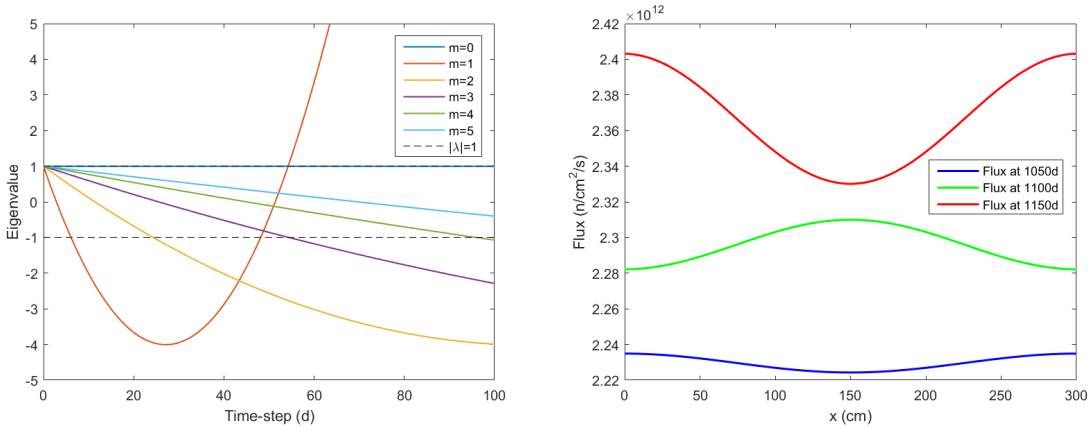


Fig. 4.5 Eigenvalues of the PC scheme with time-step when using four corrector iterations.



(a) $\alpha = 0.1$ (b) $\alpha = 0.5$

Fig. 4.6 Eigenvalues of the first instability mode of the PC scheme with time-step when using multiple corrector iterations and two different relaxation factors.



(a) Eigenvalues with time-step

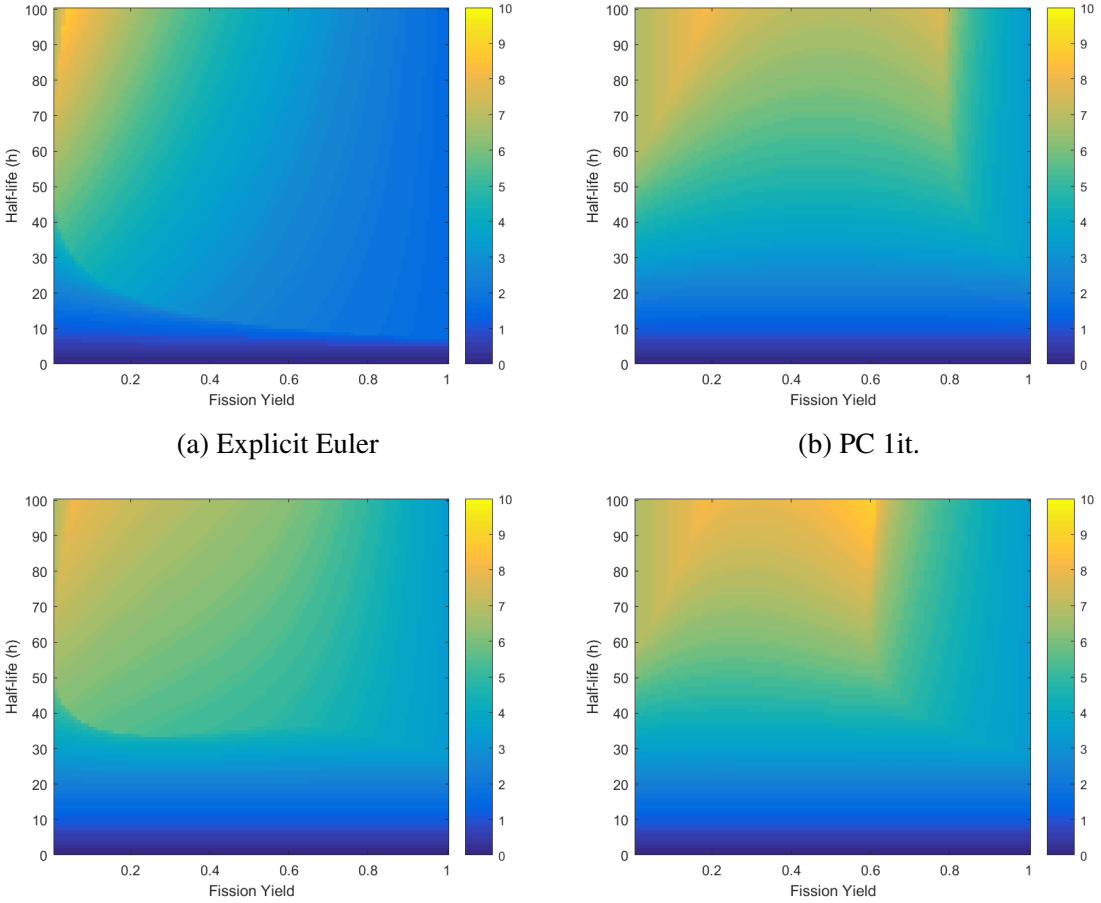
(b) Fluxes at subsequent time-points

Fig. 4.7 Excitation of a symmetric burn-up instability using a single corrector iteration and a relaxation factor of 0.1.

4.3.2 Four-nuclide system

The depletion system described above is interesting, but quite simple. The dynamics can be made more representative of a real system by including additional nuclides and decay. This adds significant complication: in the previous two-nuclide system, the two nuclide density eigenvectors represent a constant fuel material and the fissioning of the fuel into the fission product, with only the latter showing any substantial variation in its eigenvalues with time-step. With multiple fission products and decay chains, these eigenvectors cannot be so readily discarded, each with non-trivial behaviour in their eigenvalues with time-step length. For example, whereas in the two-nuclide system the zeroth nuclide density mode had a constant eigenvalue equal to one for both eigenvectors at all times, decay chains induce eigenvalues which vary substantially with time. This is not necessarily non-physical, but complicates the presentation and interpretation of results. What's more, depending on the values of fission yields and decay constants, one can obtain complex conjugate eigenpairs – the threshold for their appearance is not clear, although they occur more frequently when a decaying nuclide has a higher fission yield and shorter half-life. Again, there is nothing necessarily non-physical about this, unless the magnitude of any complex eigenvalue is greater than one.

Only a small change is made to the previous depletion system: two nuclides are added, one which is a fission product that doesn't interact with neutrons, and the other which is the decay daughter of the first, possessing a capture cross-section of 300 b. Both the half-life of the first nuclide and its fission yield are varied to examine their effect on the maximum stable time-step length. The map of maximum stable time-steps is plotted in Fig. 4.8 for the explicit



(a) Explicit Euler

(b) PC 1it.

(c) PC 2it.

(d) PC 3it.

Fig. 4.8 Maximum stable time-step length in days for different depletion schemes when varying the half-life and fission yield

Euler scheme and the PC scheme with one, two, and three corrector iterations. Again, the predicted results agree well with direct numerical simulations using various time-step lengths. There are two points to emphasise: first, as opposed to the two-nuclide system where the stable time-step lengths across schemes were identical without relaxation, each scheme now has substantially different stability properties, depending on the depletion system. This might partially explain the observation in Chapter 3 where performing an odd number of corrector iterations produced stable results while even numbers resulted in noticeable oscillation. The second point is that, for a given scheme, stability can be significantly sensitive to the decay constants and fission yield of a particular nuclide in a manner which is not readily predictable without a stability analysis.

4.4 Summary

This chapter has extended the previous stability analysis performed by Densmore et al. [24] to include the predictor-corrector method and its iterated versions. For a simple, two-nuclide system, this reproduced some of the observations previously seen in this thesis and elsewhere. These are that an odd number of corrector steps causes the flux to tilt and amplify, whereas an even number causes the flux to oscillate, and iteration with relaxation stabilises the neutronics-depletion system. With a two-nuclide system, each scheme – without relaxation – possessed the same unstable time-step length with the eigenvalues of spurious solutions growing faster with additional corrector steps. The latter phenomenon was anticipated for PC methods, but neither was expected with reference to the explicit Euler scheme – in all experiments conducted in this thesis, the explicit Euler scheme appeared to become unstable more quickly than PC schemes. This is partly explained by the simplicity of the depletion system considered: examining a four-nuclide system, the stability properties of the explicit Euler and iterated PC schemes differ substantially, depending on the values of the fission yield and decay constant. These results may also partially explain the phenomenon observed in Chapter 3 where xenon equilibrium ensured a stable solution, but only when using an odd number of corrector steps. Notably, however, none of these results explain how noise, clustering, and correlation can affect the stability of the depletion system – if the results of this chapter are definitive, their effects should be damped when taking sufficiently short time-steps. Of course, this was not observed in the extremely short time-step studies discussed in Chapter 2 – this implies that there is a degree of non-linearity affecting the stability of realistic depletion systems, at least where xenon equilibrium is not enforced. This is discussed further in Chapter 5.

Chapter 5

Conclusions and future work

5.1 Summary

This work has systematically investigated non-physical behaviour in Monte Carlo neutronics and depletion simulations. First, a gap in previous studies was addressed by including the Monte Carlo solver settings as parameters when probing instabilities. With reference to the recently revealed problem of neutron clustering, the Monte Carlo settings used in previous studies were examined and found to be inadequate to mitigate clustering effects. Numerical investigations subsequently showed that correlation and clustering are detrimental to burn-up stability and must be accounted for to achieve physical results. Likewise, clustering was shown to have frustrated previous attempts at short time-step burn-up solutions. Finally, it seems to be primarily responsible for Isotalo's hypothesised 'gadolinium instability'. Hence, this work recommends using relatively many particles per generation over fewer generations in performing Monte Carlo burn-up calculations – this is in agreement with best practices suggested by [10] and [42].

The second major study in this thesis considered variations on the predictor-corrector scheme and how they affect stability. For the standard CE/LI and two more accurate variations, the corrector step was iterated – it was shown that this, alone, does not aid stability, necessitating the use of a relaxation factor during the iteration. It was noted that the stochastic approximation is not necessarily optimal in terms of computational efficiency. This was compared against using several different fixed relaxation factors on each of the PC variants, also varying the number of corrector iterations. It was shown that a well-chosen fixed relaxation factor is more computationally efficient than the stochastic approximation in that it can achieve a stable solution in fewer corrector iterations. Using a fixed relaxation factor also seems to have the same benefit as the stochastic approximation, in that reduced settings can be applied to the corrector step transport solutions without apparent detriment to

stability. The fixed relaxation factor also seems better able to handle more correlated transport solutions than the stochastic approximation. Ultimately, the proposed fixed relaxation scheme appears to be the superior implicit scheme according to these investigations. In addition, this chapter investigated how an iterated and relaxed corrector step behaves when enforcing xenon equilibrium. Surprisingly, for the problem and time-step lengths considered, enforcing xenon equilibrium was unstable, but only if an odd number of corrector iterations were performed. This might prove significant when, say, iterating between corrector depletion, neutronics, and thermal-hydraulics during multi-physics simulations. Applying light relaxation resolved this discrepancy.

Finally, previous work exploring the numerical stability of neutronics coupled with depletion was extended to examine the predictor-corrector method and its variations with multiple corrector iterations and relaxation. The predictions were broadly in line with previous observations for a simple case. For large time-steps, odd-corrector schemes have positive eigenvalues describing the growth of their Fourier modes, resulting in a 'tilting' behaviour, while even-corrector schemes have negative eigenvalues, resulting in oscillation. The unstable time-step length for different schemes are identical without relaxation, although the magnitude of their eigenvalues increases with iteration, justifying the apparent instability of performing more corrector iterations. Relaxation on the corrector step does indeed increase stability, provided sufficient iterations are performed, although the possibility of even non-physical solution modes emerges, necessitating care in the detection of non-physical behaviour. Adding complexity to the simple depletion problem using a second, decaying fission product differentiated the time discretisation schemes further: no longer do all schemes have the same stable time-step length, with the explicit Euler and variants on the predictor-corrector scheme displaying significantly different stability behaviour, with none guaranteed superior in all cases, and with strong dependence on the characteristics of the depletion system.

5.2 The nature of burn-up instabilities

A natural question at the conclusion of this work is: what are the precise causes of burn-up instabilities? In the author's opinion, some previous works have complicated this understanding by being somewhat self-contradictory – for example, asserting that the phenomenon is not driven by noise while simultaneously introducing the stochastic approximation, which was intended for root-finding in the presence of significant noise. One of the achievements of this thesis is having sufficient results to propose a reasonably consistent explanation behind burn-up instabilities.

First, unsurprisingly, numerical instabilities can be driven by the time-step length, coupling scheme, problem geometry, and properties of the transport-depletion system. This is seen in the work of Densmore et al. [24] and in Chapter 4 – this is prior to considering a large, realistic depletion system or any inconvenient aspects of a given transport solver. This agrees with previous assertions that stability depends on the dominance ratio, as was made explicit in Chapter 4. If, however, it were as simple as that, then Chapter 2 would not exist: this is because the stability analysis performed was linear, and not all instabilities can assuredly be described by a linear analysis.

In numerical methods, a linearly stable system will damp perturbations across time-steps, preventing them from growing. Linear stability and absolute stability are not synonymous, however. By neglecting non-linear terms, linear stability analysis is implicitly restricted to dealing with perturbations of a small amplitude. If these perturbations are not sufficiently small, then a problem might be linearly stable and non-linearly unstable. Although not completely analogous (due to discrete time in numerical methods), a similar, common problem occurs in fluid dynamics, where certain flows are predicted to be laminar up to high values of the Reynolds number, whereas experiments have them transitioning to turbulence at much lower values due to finite amplitude perturbations imposed by experimental imperfections [33].

The best example or hint that linear stability may break down in this thesis is in Chapter 2 when considering a burn-up problem taking short time-steps of one hour. Without accounting for clustering and correlation, the problem falls into sustained oscillation – otherwise it is apparently stable even after very many time-steps. Although a stability analysis has not been performed, it seems that, if the problem were linearly unstable, all of the solutions would eventually have become non-physical, as all solutions possess noise that could be amplified. However, the only solution which remains physical is that in which correlation and clustering have been reduced, i.e., that in which the perturbation introduced by the Monte Carlo solver is least.

In the language of non-linear dynamics, it appears as if the transport-depletion system is subject to a sub-critical Hopf bifurcation, or rather the discrete counterpart, a subcritical Neimark-Sacker bifurcation. In the context of a depleting PWR pin, this might imply that there are two possible trajectories that a system may follow – one with a uniform burn-up, the other oscillatory, depending on the time-step length. Below some critical time-step, only the uniform trajectory is feasible – this appears to be a very short step, given oscillations were observed by Isotalo et al. [50] when taking 15 minute steps. Above this critical time-step, both trajectories for the system are possible, and the system can transition from one to the other if subject to a sufficiently large perturbation, such as noise or bias from the

transport solver. This has not been robustly demonstrated during this thesis, but all behaviours observed appear to be consistent with this hypothesis – at least where xenon equilibrium is not enforced.

Perhaps to be expected, this numerical instability does have an analogy with physical xenon oscillations which, as mentioned previously, are known to be possible and have occurred in thermal reactors [6]. Although quite sophisticated stability analyses have previously been performed [54], to the author's knowledge, none consider discrete time systems with quasi-static neutronics, which is the appropriate model to examine numerical stability. A similar physical phenomenon has long been known in thermoacoustics, where linearly stable systems can fall into self-sustained oscillation due to background noise – a process known as triggering [53].

Presently, it remains to be resolved precisely what amplitude or form of perturbation is sufficient to initiate non-physical behaviour in burn-up simulations. Undoubtedly, Monte Carlo noise, correlation, bias, or even a lack of convergence could all be detrimental to stability and should be avoided – but what lengths must one go to in order to rule out their effects? Until this can be reasonably approximated, a linear stability criterion may be of little practical use. For example, although efforts were made to minimise clustering and correlation, it is difficult to assert that the simulations shown in Fig. 3.1 became unstable due to violating either a linear or non-linear stability criterion. Chapter 2 demonstrates that minimising correlation has a dramatic effect on stability, but it may only reduce the frequency with which sufficiently large amplitude perturbations are introduced, rather than removing them altogether. The critical size of perturbation is also likely to depend upon the time-step taken.

This appears to be the success of enforcing xenon equilibrium: it removes this prominent non-linear instability (while also, likely, improving linear stability) by modifying the depletion system slightly. It does not, however, provide any assurance of stability beyond this; depending on the scheme, time-step length, or (in relatively extreme cases) transport solver settings, non-physical behaviour can re-emerge, as seen towards the end of Chapters 2 and 3.

As is the case across computational physics, an alternative, more universal means of achieving a stable solution is to use an implicit time-stepping scheme. The drawbacks of implicit methods are also universal: they impose greater computational expense than their explicit counterparts, and, if using a relaxation method, the choice can be difficult without prior experience.

5.3 Future work

5.3.1 Burn-up stability diagnosis

Identifying when non-physical solutions will or are likely to occur remains challenging, except in hindsight. The ability to do so would allow for more optimal usage of computational resources: new problems would not require lengthy sensitivity studies, time-steps could be chosen to be as long as feasible, and implicit schemes could be applied only where necessary. The most promising beginning of such work has been introduced by Densmore et al. in estimating the stable time-step length for a simple deterministic system [24], with an extension given by Chapter 4.

However, this area is far from developed. Higher-order methods are yet to be considered and are likely to have different stability properties, the complexity of the depletion system could be increased to be more representative of the hundreds-to-thousands of isotopes commonly included in burn-up calculations, and, further, the diffusion approximation might be relaxed with a simple transport model in 1D. Even if the analysis is intractable to perform on-the-fly as a stability metric, with sufficient complexity it may allow for the development of time-step selection heuristics for various problem dimensions or power normalisations.

The first significant complication, as mentioned above, is the necessity of considering non-linear effects. Monte Carlo solvers introduce perturbations to the neutronics-depletion system, and it is not trivial to decide whether or not these will be detrimental to stability. In continuous-time dynamical systems, a weakly non-linear analysis is often applied in such cases, typically using the long-established method of multiple time-scales [53, 54]. This method is explicitly only applicable to continuous time systems, whereas numerical schemes are, of course, in discrete time. Hence, some discrete analogue of this technique must be applied.

Using a deterministic solver, however, may obviate this particular concern by removing the large perturbations introduced by Monte Carlo which may trigger instability. Indeed, a direct comparison between Monte Carlo and a high-fidelity deterministic solver may be an efficient means of disentangling the role of noise and correlation.

Nevertheless, in many cases coupling Monte Carlo and depletion may not be limited by numerical stability – in the presence of strong burnable absorbers, time-steps may be constrained by accuracy, at least initially. This concern may be partially alleviated by the wider implementation of higher-order methods.

5.3.2 Newton iteration

A solution technique which might be applied to stabilising neutronics and depletion is Newton's method, or the Newton iteration, familiar across numerical analysis as a root-finding method for solving equations of the form:

$$H(x) = 0 \tag{5.1}$$

The non-linear system given by Eq. (3.8) can be recast in this form by subtracting the input vector such that, at a fixed point:

$$F(N^*_{n+1}) - N^*_{n+1} = 0 \tag{5.2}$$

Hence, Newton iteration can be applied to locating this fixed point by defining a modified non-linear system:

$$G(N^{(k)}_{n+1}) := F(N^{(k)}_{n+1}) - N^{(k)}_{n+1} \tag{5.3}$$

Each iteration of Newton's method proceeds from some initial guess ($N^{(k)}_{n+1}$ here) by evaluating the non-linear function – performing a corrector iteration. However, in addition to this, gradient information must be computed – the Jacobian matrix at $N^{(k)}_{n+1}$ must be evaluated. Note that this matrix is the same as that in Eq. (3.11) but with an identity matrix subtracted. Obtaining the Jacobian allows for the following Taylor expansion about $N^{(k)}_{n+1}$:

$$G(N^*_{n+1}) = 0 \approx G(N^{(k)}_{n+1}) + J^{(k)}_G \cdot \delta N^{(k)}_{n+1} \tag{5.4}$$

The linear system obtained by setting the right-hand side of Eq. (5.4) exactly equal to the zero vector can be solved for $\delta N^{(k)}_{n+1}$, the update vector which gives $N^{(k+1)}_{n+1}$ as:

$$N^{(k+1)}_{n+1} = N^{(k)}_{n+1} + \delta N^{(k)}_{n+1} \tag{5.5}$$

For general non-linear systems, Newton's method, if initialised with a sufficiently good guess, can be shown to converge quadratically towards the fixed point. However, there are a number of difficulties associated with the method:

- Choosing a sufficiently good starting point for rapid convergence
- Obtaining gradient information is often expensive
- Large Jacobian matrices are expensive to invert
- Large Jacobian matrices may have a prohibitive memory footprint

These problems may be partially circumvented or justified through other benefits. For example, one significant advantage which might be realised with a Newton method is the efficient solution of a multi-physics problem in few iterations. That is, the nuclide density distribution as well as, say, the thermal-hydraulic field might be solved for simultaneously, rather than requiring an iteration between three separate physics solvers which would otherwise be necessary.

A Newton-based LE/LI scheme is shown in Algorithm 8.

Input: $N_0, n_{\text{steps}}, \Delta t, n_{\text{ss}}, k_{\max}$
for $n = 0, \ldots, n_{\text{steps}} - 1$ **do**
  $\mathbf{A}_n \leftarrow \psi(N_n)$
  if $n > 0$ **then**
    $N \leftarrow N_n$
    for $s = 0, \ldots, n_{\text{ss}} - 1$ **do**
      $\mathbf{A} \leftarrow w_{n,s}\mathbf{A}_n + w_{n-1,s}\mathbf{A}_{n-1}$
      $N \leftarrow \exp\left[\mathbf{A}\frac{\Delta t_n}{n_{\text{ss}}}\right] N$
    end
    $N_{n+1}^{(0)} \leftarrow N$
  else
    $N_{n+1}^{(0)} \leftarrow \exp\left[\mathbf{A}_n \Delta t_n\right] N_n$
  end
  for $k = 0, \ldots, k_{\max} - 1$ **do**
    $\left[\mathbf{A}_{n+1}^{(k)}, J_F^{(k)}\right] \leftarrow \psi(N_{n+1}^{(k)})$
    $\bar{N} \leftarrow N_n$
    for $s = 0, \ldots, n_{\text{ss}} - 1$ **do**
      $\mathbf{A} \leftarrow w_{n,s}\mathbf{A}_n + w_{n+1,s}\mathbf{A}_{n+1}^{(k)}$
      $N \leftarrow \exp\left[\mathbf{A}\frac{\Delta t_n}{n_{\text{ss}}}\right] N$
    end
    $\bar{N}_{n+1} \leftarrow N$
    $\delta N \leftarrow \left(I - J_F^{(k)}\right)^{-1} \left(\bar{N}_{n+1} - N_{n+1}^{(k)}\right)$
    $N_{n+1}^{(k+1)} \leftarrow N_{n+1}^{(k)} + \delta N$
  end
  $N_{n+1} \leftarrow N_{n+1}^{(k_{\max})}$
end

Algorithm 8: LE/LI substep method with Newton iteration

One of the largest complexities in Newton's method is generating the Jacobian matrix, J_F. It is possible, in principle, to obtain the Jacobian by finite differencing, although due to

both the computational expense of a corrector iteration and the noise associated with Monte Carlo estimates, such an effort would be entirely impractical.

Alternatively, using the chain rule and recalling the definition of F from Eq. (3.9), the Jacobian can be decomposed into two separate matrices. Referring to the Jacobian of $F(N_{n+1})$ which can easily be converted to J_G, one has:

$$J_F = \frac{\partial F(N_{n+1})}{\partial N_{n+1}} = J_\beta \cdot J_\tau = \frac{\partial \beta(\psi_{\text{EOS}})}{\partial \psi_{\text{EOS}}} \cdot \frac{\partial \tau(N_{n+1})}{\partial N_{n+1}} \tag{5.6}$$

Here J_β and J_τ are the depletion and neutronics Jacobians, respectively. The Newton iteration would require computing both of these matrices. A similar logic was followed by Aufiero and Fratoni, but applied to coupling Monte Carlo with thermal-hydraulics [2].

One could calculate the depletion matrix relatively straightforwardly: given that a Bateman solution is both deterministic and relatively inexpensive to evaluate as compared to a transport solve, this Jacobian can be obtained using finite differences. That is, for a change in the i-th element of the neutronics field, $\psi_{\text{EOS},i}$, the depletion derivative can be approximated as:

$$\frac{\partial \beta(\psi_{\text{EOS}})}{\partial \psi_{\text{EOS},i}} \approx \frac{\beta(\psi_{\text{EOS}} + \delta\psi_{\text{EOS},i}) - \beta(\psi_{\text{EOS}} - \delta\psi_{\text{EOS},i})}{2\delta\psi_{\text{EOS},i}} \tag{5.7}$$

This can be applied to all one-group cross-sections and fluxes of interest without incurring significant computational expense – at least for problems on the scale of an assembly. Finite differencing can also be applied identically to all variations of corrector step – the use of substeps and linear or quadratic interpolation would be accounted for within the evaluation of β.

The transport derivatives, however, are much more difficult to obtain: as mentioned previously, finite differencing is ruled out both due to the computational expense of Monte Carlo solutions, as well as due to the stochastic noise associated with results which may contaminate the estimation of derivatives. Instead, one can apply Monte Carlo perturbation theory to efficiently calculate the necessary derivatives. The relevant details of perturbation theory will not be elaborated upon here, but for a good introduction to Generalised Perturbation Theory, the beginning of the paper by Cacuci [13] is a good reference, while Monte Carlo perturbation theory is well described by Kiedrowski [57].

From perturbation theory, it is possible to obtain the derivatives of quantities which feature in the burn-up matrix with respect to a given nuclide density. For example, a one-group capture cross-section of isotope i in burnable region b, as a response function, would be:

$$R = \sigma_{\text{c},i,b} = \frac{\langle \sigma_{\text{c},i}, \psi \rangle_b}{\langle 1, \psi \rangle_b} \tag{5.8}$$

Here b is used as a sub-script following the ket to denote a restriction on the domain of integration – namely to the spatial domain of burnable region b. If the density of nuclide j in region d is perturbed, the sensitivity of $\sigma_{\mathrm{c},i,b}$ to nuclide j is:

$$S_N^{\sigma} = \frac{\langle N_{j,d}\frac{\partial \sigma_{\mathrm{c},i,b}}{\partial N_{j,d}}, \psi\rangle_b}{\langle \sigma_{\mathrm{c},i}, \psi\rangle_b} - \frac{\langle N_{j,d}\frac{\partial 1}{\partial N_{j,d}}, \psi\rangle_b}{\langle 1, \psi\rangle_b} + \frac{\langle \sigma_{\mathrm{c},i}, N_{j,d}\frac{\partial \psi}{\partial N_{j,d}}\rangle_b}{\langle \sigma_{\mathrm{c},i}, \psi\rangle_b} - \frac{\langle 1, N_{j,d}\frac{\partial \psi}{\partial N_{j,d}}\rangle_b}{\langle 1, \psi\rangle_b} \tag{5.9}$$

The latter two terms are the indirect terms – these may be calculated using sensitivity estimators. The first two terms are direct – in this case both values are trivially zero given that a change in a nuclide density does not affect the value of a microscopic cross-section and the derivative of a constant is zero. Hence, the sensitivity of a one-group cross-section to a nuclide density is:

$$S_N^{\sigma} = \frac{\langle \sigma_{\mathrm{c},i}, N_{j,d}\frac{\partial \psi}{\partial N_{j,d}}\rangle_b}{\langle \sigma_{\mathrm{c},i}, \psi\rangle_b} - \frac{\langle 1, N_{j,d}\frac{\partial \psi}{\partial N_{j,d}}\rangle_b}{\langle 1, \psi\rangle_b} \tag{5.10}$$

A similar approach can be taken to calculate the derivative of the flux amplitude, ϕ, in burnable region b. The response function for this is:

$$R = \phi_b = P_0 \frac{\langle 1, \psi\rangle_b}{\langle E_{\mathrm{fiss}}\Sigma_{\mathrm{f}}, \psi\rangle} \tag{5.11}$$

In this equation, P_0 is the power of the system and E_{fiss} is the energy released per fission. Given that the energy released per fission varies with the isotope being fissioned, the notation here is simplified by taking:

$$E_{\mathrm{fiss}}\Sigma_f = \sum_{k}^{\text{isotopes}} E_{\mathrm{fiss},k}\sigma_{f,k}N_k \tag{5.12}$$

with $E_{\mathrm{fiss},k}$ as the energy released from fissioning isotope k. The denominator of the equation is the total power of the system and is necessary to ensure that the system is normalised to a constant power, even when subject to a perturbation. For a perturbation in the same nuclide density as above ($N_{j,d}$), the sensitivity of ϕ_b is:

$$S_N^{\phi} = \frac{\langle N_{j,d}\frac{\partial 1}{\partial N_{j,d}}, \psi\rangle_b}{\langle 1, \psi\rangle_b} - \frac{\langle N_{j,d}\frac{\partial E_{\mathrm{fiss}}\Sigma_{\mathrm{f}}}{\partial N_{j,d}}, \psi\rangle}{\langle E_{\mathrm{fiss}}\Sigma_{\mathrm{f}}, \psi\rangle} + \frac{\langle 1, N_{j,d}\frac{\partial \psi}{\partial N_{j,d}}\rangle_b}{\langle 1, \psi\rangle_b} - \frac{\langle E_{\mathrm{fiss}}\Sigma_{\mathrm{f}}, N_{j,d}\frac{\partial \psi}{\partial N_{j,d}}\rangle}{\langle E_{\mathrm{fiss}}\Sigma_{\mathrm{f}}, \psi\rangle} \tag{5.13}$$

As before, the latter two terms are indirect and can be calculated using sensitivity capabilities. Of the former two terms, the first is zero through differentiating a constant. However, the

latter is non-negligible, but, using Eq. (5.12), reduces to:

$$\frac{\langle N_{j,d}\frac{\partial E_{\text{fiss}}\Sigma_{\text{f}}}{\partial N_{j,d}},\psi\rangle}{\langle E_{\text{fiss}}\Sigma_{\text{f}},\psi\rangle} = \frac{\langle E_{\text{fiss},j}\Sigma_{\text{f},j},\psi\rangle_d}{\langle E_{\text{fiss}}\Sigma_{\text{f}},\psi\rangle} \tag{5.14}$$

This term is simply the fraction of total power produced by the fission of isotope j in region d, easily calculated during a standard transport simulation. Therefore, the sensitivity coefficient for a one-group flux perturbed by a change in a nuclide density is given by:

$$S_N^{\phi} = -\frac{\langle E_{\text{fiss},j}\Sigma_{\text{f},j},\psi\rangle_d}{\langle E_{\text{fiss}}\Sigma_{\text{f}},\psi\rangle} + \frac{\langle 1, N_{j,d}\frac{\partial\psi}{\partial N_{j,d}}\rangle_b}{\langle 1,\psi\rangle_b} - \frac{\langle E_{\text{fiss}}\Sigma_{\text{f}}, N_{j,d}\frac{\partial\psi}{\partial N_{j,d}}\rangle}{\langle E_{\text{fiss}}\Sigma_{\text{f}},\psi\rangle} \tag{5.15}$$

Given that fission yields vary relatively weakly with changes in flux spectrum, the sensitivity expressions given by Eqs. (5.10) and (5.15) – when converted to derivatives – allow for the computation of all elements of the transport Jacobian, J_τ.

During the course of this thesis, Newton's method was briefly pursued but suffered from a number of difficulties: estimating many sensitivity coefficients is computationally challenging both in terms of speed and memory, the sensitivity coefficients obtained will be affected by stochastic noise, and one must choose which responses and perturbations to compute. In practice, this rendered the application of Newton's method to depletion nearly infeasible. Although there were some hints of improved stability, the computational burden could not justify the application of the method in lieu of relaxation. However, this is not to entirely rule it out; following from Aufiero and Fratoni [2], the number of depletion sensitivities to calculate might be substantially reduced by representing the depletion problem with some relatively low order functional expansion and obtaining sensitivities with respect to these [3].

5.3.3 In-line depletion

A perhaps more scalable, 'Monte Carlo-esque' approach might be deployed for stable depletion coupling. The treatment of the corrector-step in Chapter 3 is analogous with work in coupling Monte Carlo neutronics with thermal-hydraulics, i.e., performing fixed-point iteration with relaxation to ensure a stable solution. This analogy suggests an extension to the present work: Gill et al. [36] describe an 'in-line' method for converging thermal-hydraulics and neutronics, similar to that used when enforcing xenon equilibrium with Monte Carlo [38, 50]. This method has been further investigated by Kreher et al. [65]. Their in-line thermal-hydraulics algorithm consists of performing thermal-hydraulic updates during the inactive cycles of the Monte Carlo simulation, exchanging power and thermal-hydraulic

information between the two solvers prior to fission source convergence – this requires sufficient histories to be simulated during the data exchanges to ensure statistical noise is not detrimental to the results, of course. Although there has been little in the way of mathematical analysis to support this strategy, numerical experiments find its performance to be comparable to applying a relaxation [36, 65].

An analogous coupled solution strategy exists for deterministic solvers with promising results and some analysis. Senecal & Ji [88] introduced the concept of a cost of 'over-solving' in multi-physics simulations: it is not desirable during an iterate between separate physics solvers to tightly converge each solution before exchanging information between the solvers – much of the computational effort will be wasted obtaining precision on problems with essentially erroneous boundary conditions. Converging each solver less tightly before passing their solution to the other solver showed a significant improvement in performance. Applied to nuclear engineering, Shen et al. [89] highlight that, when iterating between the CMFD solver in MPACT and thermal-hydraulics, the problem converges more rapidly when the CMFD is less tightly solved. They also perform a Fourier/Von Neumann stability analysis to demonstrate the stabilising effects of such a strategy are more optimal than using a fixed relaxation factor.

Given that the stable corrector step nuclide density is given by the fixed-point of a discrete map, an identical scheme could be applied to depletion. During the inactive cycles of the EOS transport solution, reactions rates could be accumulated for a chosen number of cycles before performing a 'depletion update', i.e., evaluating a corrector step Bateman equation solution and updating the EOS nuclide densities accordingly. As discussed by Gill et al. and Kreher et al., depending on the degree of statistical contamination between updates, this should provide some equivalence to a relaxation. While Gill et al. point out that this strategy is somewhat antithetical to the use of fission source convergence techniques, it is possible that it might allow for a more efficient PC scheme implementation wherein two separate transport solutions are no longer necessary: the corrector step takes places only during the inactive cycles, the predictor step uses the reaction rates accumulated by the conclusion of the active cycles, obviating the second full transport solution at each time-step. The application of fission source acceleration might even prove to be limited in the case of realistic, whole-core depletion problems: often for reactor analysis, at a given time-point, one might typically also apply either a critical boron or control rod position search, the usual implementation of which is carried out over many inactive cycles. In principle, alongside in-line thermal-hydraulic feedback, the proposed in-line depletion schemes might also be carried out simultaneously.

It should be emphasised that this algorithm is substantially different from those used to enforce xenon equilibrium: here the discrete map being iterated over is that given by

Eq. (3.9), whereas xenon equilibrium assumes xenon and iodine have no time dependence, and is given by a map derived from their equilibrium Eqs. (1.24) and (1.25).

Unfortunately, such a scheme requires tight coupling between the Monte Carlo solver and depletion engine to exchange information between the two while holding tallies and the fission source in memory during the depletion calculations, and modifying isotopic compositions between cycles during the transport solution. It is impractical to modify a Monte Carlo code to allow such a coupling within the time-frame of the present work, but this appears to be a promising avenue for future research.

References

[1] Askew, J. (1972). A Characteristics Formulation of the Neutron Transport Equation in Complicated Geometries. Technical Report AEEW–M1108, UK Atomic Energy Establishment.

[2] Aufiero, M. and Fratoni, M. (2017). A new approach to the stabilization and convergence acceleration in coupled Monte Carlo–CFD calculations: The Newton method via Monte Carlo perturbation theory. *Nuclear Engineering and Technology*, 49(6):1181–1188.

[3] Aufiero, M., Martin, M., and Fratoni, M. (2016). XGPT: Extending Monte Carlo Generalized Perturbation Theory capabilities to continuous-energy sensitivity functions. *Annals of Nuclear Energy*, 96:295–306.

[4] Bahadir, T. and Lindahl, S. (2009). Studsvik's next generation nodal code SIMULATE-5. In *Proc. Advanced in Nuclear Fuel Management IV*, Hilton Head Island, South Carolina.

[5] Battistoni, G., Boehlen, T., Cerutti, F., Chin, P. W., Esposito, L. S., Fassò, A., Ferrari, A., Lechner, A., Empl, A., Mairani, A., Mereghetti, A., Ortega, P. G., Ranft, J., Roesler, S., Sala, P. R., Vlachoudis, V., and Smirnov, G. (2015). Overview of the FLUKA code. *Annals of Nuclear Energy*, 82:10–18.

[6] Bell, G. and Glasstone, S. (1970). *Nuclear Reactor Theory*. US Atomic Energy Commission, Washington, DC.

[7] Bird, G. A. (1994). *Molecular gas dynamics and the direct simulation of gas flows*. Clarendon Press, Oxford, 2nd edition.

[8] Boyd, W., Shaner, S., Li, L., Forget, B., and Smith, K. (2014). The OpenMOC method of characteristics neutral particle transport code. *Annals of Nuclear Energy*, 68:43–52.

[9] Brissenden, R. J. and Garlick, A. R. (1986). Biases in the estimation of Keff and its error by Monte Carlo methods. *Annals of Nuclear Energy*, 13(2):63–83.

[10] Brown, F. B. (2011a). 'K-effective of the World' and Other Concerns for Monte Carlo Eigenvalue Calculations. *Progress in Nuclear Science and Technology*, 2:738–742.

[11] Brown, F. B. (2011b). Recent Advances and Future Prospects for Monte Carlo. *Progress in Nuclear Science and Technology*, 2:1–4.

[12] Brun, E., Dumonteil, E., Hugot, F.-X., Huot, N., Jouanne, C., Lee, Y., Malvagi, F., Mazzolo, A., Petit, O., Trama, J.-C., and Zoia, A. (2011). Overview of TRIPOLI-4 version 7, Continuous-energy Monte-Carlo Transport Code. In *Proc. ICAPP*.

[13] Cacuci, D. G. (2017). A paradigm-shifting methodology for sensitivity analysis of critical multiplying nuclear systems. *Nuclear Science and Engineering*, 185(3):361–383.

[14] Carmona, R. A., Del Moral, P., Hu, P., and Oudjane, N. (2012). Numerical Methods in Finance. In *Springer Proceedings in Mathematics*, volume 12.

[15] Carney, S., Brown, F., Kiedrowski, B., and Martin, W. (2014). Theory and applications of the fission matrix method for continuous-energy Monte Carlo. *Annals of Nuclear Energy*, 73:423–431.

[16] Carpenter, D. and Wolf III, J. H. (2010). The log linear rate constant power depletion method. In *Proc. PHYSOR 2010*, Pittsburgh, Pennsylvania.

[17] Chadwick, M. B., Herman, M., Obložinský, P., Dunn, M. E., Danon, Y., Kahler, A. C., Smith, D. L., Pritychenko, B., Arbanas, G., Arcilla, R., Brewer, R., Brown, D. A., Capote, R., Carlson, A. D., Cho, Y. S., Derrien, H., Guber, K., Hale, G. M., Hoblit, S., Holloway, S., Johnson, T. D., Kawano, T., Kiedrowski, B. C., Kim, H., Kunieda, S., Larson, N. M., Leal, L., Lestone, J. P., Little, R. C., McCutchan, E. A., MacFarlane, R. E., MacInnes, M., Mattoon, C. M., McKnight, R. D., Mughabghab, S. F., Nobre, G. P., Palmiotti, G., Palumbo, A., Pigni, M. T., Pronyaev, V. G., Sayer, R. O., Sonzogni, A. A., Summers, N. C., Talou, P., Thompson, I. J., Trkov, A., Vogt, R. L., van der Marck, S. C., Wallner, A., White, M. C., Wiarda, D., and Young, P. G. (2011). ENDF/B-VII.1 nuclear data for science and technology: Cross sections, covariances, fission product yields and decay data. *Nuclear Data Sheets*, 112(12):2887–2996.

[18] Cochet, B., Jinaphanh, A., Heulers, L., and Jacquet, O. (2015). Capabilities overview of the MORET 5 Monte Carlo code. *Annals of Nuclear Energy*, 82:74–84.

[19] Cosgrove, P., Shwageraus, E., and Parks, G. T. (2020a). A simple implicit coupling scheme for Monte Carlo neutronics and isotopic depletion. *Annals of Nuclear Energy*, 141.

[20] Cosgrove, P., Shwageraus, E., and Parks, G. T. (2020b). Neutron clustering as a driver of Monte Carlo burn-up instability. *Annals of Nuclear Energy*, 137.

[21] Croff, A. (1980). A User's Manual For The ORIGEN2 Computer Code. Technical report, Oak Ridge National Laboratory.

[22] De Mulatier, C., Dumonteil, E., Rosso, A., and Zoia, A. (2015). The critical catastrophe revisited. *Journal of Statistical Mechanics: Theory and Experiment*, 2015(8):P08021.

[23] Demazière, C. (2019). *Modelling of Nuclear Reactor Multi-physics*. Academic Press.

[24] Densmore, J. D., Gill, D. F., and Griesheimer, D. P. (2013). Stability analysis of burnup calculations. *Transactions of the American Nuclear Society*, 109(2012):695–698.

[25] Duderstadt, J. J. and Hamilton, L. J. (1976). *Nuclear Reactor Analysis*. John Wiley and Sons, Inc., New York.

[26] Dufek, J. and Hoogenboom, J. E. (2009). Numerical stability of existing Monte Carlo burnup codes in cycle calculations of critical reactors. *Nuclear Science and Engineering*, 162(3):307–311.

[27] Dufek, J., Kotlyar, D., and Shwageraus, E. (2013a). The stochastic implicit Euler method: A stable coupling scheme for Monte Carlo burnup calculations. *Annals of Nuclear Energy*, 60:295–300.

[28] Dufek, J., Kotlyar, D., Shwageraus, E., and Leppänen, J. (2013b). Numerical stability of the predictor-corrector method in Monte Carlo burnup calculations of critical reactors. *Annals of Nuclear Energy*, 56:34–38.

[29] Dumonteil, E., Bruna, G., Malvagi, F., Onillon, A., and Richet, Y. (2017). Clustering and traveling waves in the Monte Carlo criticality simulation of decoupled and confined media. *Nuclear Engineering and Technology*, 49(6):1157–1164.

[30] Dumonteil, E. and Courau, T. (2010). Dominance ratio assessment and Monte Carlo criticality simulations: Dealing with high dominance ratio systems. *Nuclear Technology*, 172(2):120–131.

[31] Dumonteil, E. and Diop, C. M. (2011). Biases and Statistical Errors in Monte Carlo Burnup Calculations: An Unbiased Stochastic Scheme to Solve Boltzmann/Bateman Coupled Equations. *Nuclear Science and Engineering*, 167(2):165–170.

[32] Dumonteil, E., Malvagi, F., Zoia, A., Mazzolo, A., Artusio, D., Dieudonné, C., and De Mulatier, C. (2014). Particle clustering in Monte Carlo criticality simulations. *Annals of Nuclear Energy*, 63:612–618.

[33] Eckhardt, B., Schneider, T. M., Hof, B., and Westerweel, J. (2007). Turbulence Transition in Pipe Flow. *Annual Review of Fluid Mechanics*, 39(1):447–468.

[34] Ellis, M., Josey, C., Forget, B., and Smith, K. (2016). Spatially Continuous Depletion Algorithm for Monte Carlo Simulations. *Transactions of the American Nuclear Society*, 115.

[35] Fridman, E., Shwageraus, E., and Galperin, A. (2008). Efficient generation of one-group cross sections for coupled Monte Carlo depletion calculations. *Nuclear Science and Engineering*, 159(1):37–47.

[36] Gill, D. F., Griesheimer, D. P., and Aumiller, D. L. (2017). Numerical Methods in Coupled Monte Carlo and Thermal-Hydraulic Calculations. *Nuclear Science and Engineering*, 185:194–205.

[37] Griesheimer, D., Gill, D., Nease, B., Sutton, T., Stedry, M., Dobreff, P., Carpenter, D., Trumbull, T., Caro, E., Joo, H., and Millman, D. (2015). MC21 v.6.0 – A continuous-energy Monte Carlo particle transport code with integrated reactor feedback capabilities. *Annals of Nuclear Energy*, 82:29–40.

[38] Griesheimer, D. P. (2010). In-line xenon convergence algorithm for Monte Carlo reactor calculations. In *Proc. PHYSOR 2010*, Pittsburgh, Pennsylvania.

[39] Griesheimer, D. P., Carpenter, D. C., and Stedry, M. H. (2017). Practical techniques for large-scale Monte Carlo reactor depletion calculatons[sic]. *Progress in Nuclear Energy*, 101(C):409–423.

[40] Haeck, W., Cochet, B., and Aguiar, L. (2012). A burnup-dependent isomeric production branching ratio treatment. *Nuclear Science and Engineering*, 171(1):52–68.

[41] Haeck, W. and Verboomen, B. (2007). An optimum approach to Monte Carlo burnup. *Nuclear Science and Engineering*, 156(2):180–196.

[42] Herman, B. R. (2014). *Monte Carlo and Thermal Hydraulic Coupling using Low-Order Nonlinear Diffusion Acceleration*. PhD thesis, Massachusetts Institute of Technology.

[43] Hikes, J. and Ferrer, R. (2013). Solving the Bateman Equations in CASMO5 using Implicit ODE Numerical Methods for Stiff Systems. In *Proc. M&C 2013*, Sun Valley, Idaho.

[44] Horelik, N., Herman, B. R., Forget, B., and Smith, K. (2013). Benchmark for Evaluation and Validation of Reactor Simulations (BEAVRS). In *Proc. International Conference on Mathematics, Computational Methods & Reactor Physics*, volume 4.

[45] Hutt, P. K., Gaines, N., Halsall, M. J., McEllin, M., and White, R. J. (1991). UK core performance code package. *Nuclear Energy*, 30(5):291–298.

[46] Isotalo, A. (2013). *Computational Methods for Burnup Calculations with Monte Carlo Neutronics*. PhD thesis, Aalto University.

[47] Isotalo, A. and Aarnio, P. (2011a). Comparison of depletion algorithms for large systems of nuclides. *Annals of Nuclear Energy*, 38(2-3):261–268.

[48] Isotalo, A. and Aarnio, P. (2011b). Higher order methods for burnup calculations with Bateman solutions. *Annals of Nuclear Energy*, 38(9):1987–1995.

[49] Isotalo, A. and Aarnio, P. (2011c). Substep methods for burnup calculations with Bateman solutions. *Annals of Nuclear Energy*, 38(11):2509–2514.

[50] Isotalo, A. E., Leppänen, J., and Dufek, J. (2013). Preventing xenon oscillations in Monte Carlo burnup calculations by enforcing equilibrium xenon distribution. *Annals of Nuclear Energy*, 60:78–85.

[51] Josey, C. (2017). *Development and Analysis of High Order Neutron Transport-Depletion Coupling Algorithms*. PhD thesis, Massachusetts Institute of Technology.

[52] Josey, C., Forget, B., and Smith, K. (2017). High order methods for the integration of the Bateman equations and other problems of the form of $y' = F(y,t)y$. *Journal of Computational Physics*, 350:296–313.

[53] Juniper, M. P. (2012). Triggering in Thermoacoustics. *International Journal of Spray and Combustion Dynamics*, 4(3):217–237.

[54] Kastin, N., Meron, E., Kolin, A., and Kinast, S. (2019). Nonlinear stability and limit cycles in xenon-induced reactor oscillations. *Progress in Nuclear Energy*, 116:168–179.

[55] Kępisty, G. and Cetnar, J. (2015a). Burnup instabilities in the full-core HTR model simulation. *Annals of Nuclear Energy*, 85:652–661.

[56] Kępisty, G. and Cetnar, J. (2015b). Instabilities of Monte-Carlo burnup calculations for nuclear reactors - Demonstration and dependence from time step model. *Nuclear Engineering and Design*, 286:49–59.

[57] Kiedrowski, B. C. (2017). Review of Early 21st-Century Monte Carlo Perturbation and Sensitivity Techniques for k -Eigenvalue Radiation Transport Calculations. *Nuclear Science and Engineering*, 185(3):426–444.

[58] Kochunas, B., Collins, B., Downar, T., and Martin, W. (2013). MPACT: Michigan Parallel Advanced Characteristics Transport. In *Proc. M&C 2013*, Sun Valley, Idaho.

[59] Kotlyar, D. and Shwageraus, E. (2013a). Comparison of Monte Carlo-Burnup-Thermal Hydraulic Coupling Schemes. In *American Nuclear Society Student Conference*, Boston, Massachusetts.

[60] Kotlyar, D. and Shwageraus, E. (2013b). On the use of predictor-corrector method for coupled Monte Carlo burnup codes. *Annals of Nuclear Energy*, 58:228–237.

[61] Kotlyar, D. and Shwageraus, E. (2014). Numerically stable Monte Carlo-burnup-thermal hydraulic coupling schemes. *Annals of Nuclear Energy*, 63:371–381.

[62] Kotlyar, D. and Shwageraus, E. (2016a). Stochastic semi-implicit substep method for coupled depletion Monte-Carlo codes. *Annals of Nuclear Energy*, 92:52–60.

[63] Kotlyar, D. and Shwageraus, E. (2016b). Sub-step methodology for coupled Monte Carlo depletion and thermal hydraulic codes. *Annals of Nuclear Energy*, 96:61–75.

[64] Kowalski, M. A. and Shwageraus, E. (2020). A hybrid continuous energy and multi-group Monte Carlo method. *Annals of Nuclear Energy*, 140.

[65] Kreher, M., Forget, B., and Smith, K. (2019). Single-Batch Monte Carlo Multiphysics Coupling. In *Proc. M&C 2019*, Portland, Oregon.

[66] Leppänen, J. (2019). Acceleration of fission source convergence in the Serpent 2 Monte Carlo code using a response matrix based solution for the initial source distribution. *Annals of Nuclear Energy*, 128:63–68.

[67] Leppänen, J. and Mattila, R. (2015). Study on computational performance in generation of cross sections for nodal simulators using continuous-energy Monte Carlo calculations. *Journal of Nuclear Science and Technology*, 52(7-8):945–952.

[68] Leppänen, J., Pusa, M., and Fridman, E. (2016). Overview of methodology for spatial homogenization in the Serpent 2 Monte Carlo code. *Annals of Nuclear Energy*, 96:126–136.

[69] Leppänen, J., Pusa, M., Viitanen, T., Valtavirta, V., and Kaltiaisenaho, T. (2015). The Serpent Monte Carlo code: Status, development and applications in 2013. *Annals of Nuclear Energy*, 82:142–150.

[70] Lieberoth, J. (1968). A Monte Carlo technique to solve the static eigenvalue problem of the Boltzmann transport equation. *Nukleonik*, 11:213–219.

[71] Lindley, B. A., Hosking, J. G., Smith, P. J., Powney, D. J., Tollit, B. S., Newton, T. D., Perry, R., Ware, T. C., and Smith, P. N. (2017). Current status of the reactor physics code WIMS and recent developments. *Annals of Nuclear Energy*, 102:148–157.

[72] Lux, I. and Koblinger, L. (1991). *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations*. CRC Press.

[73] Miller Jr., W. F. and Lewis, E. E. (1984). *Computational Methods of Neutron Transport*. John Wiley and Sons, Inc., New York.

[74] Moler, C. and Van Loan, C. (2003). Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later. *SIAM Review*, 45(1):3–49.

[75] Nagaya, Y., Okumura, K., and Mori, T. (2015). Recent developments of JAEA's Monte Carlo code MVP for reactor physics applications. *Annals of Nuclear Energy*, 82:85–89.

[76] Nowak, M., Miao, J., Dumonteil, E., Forget, B., Onillon, A., Smith, K. S., and Zoia, A. (2016). Monte Carlo power iteration: Entropy and spatial correlations. *Annals of Nuclear Energy*, 94:856–868.

[77] Pandya, T. M., Johnson, S. R., Evans, T. M., Davidson, G. G., Hamilton, S. P., and Godfrey, A. T. (2016). Implementation, capabilities, and benchmarking of Shift, a massively parallel Monte Carlo radiation transport code. *Journal of Computational Physics*, 308:239–272.

[78] Poston, D. and Trellue, H. (1999). Development of a Fully-Automated Monte Carlo Burnup Code Monteburns. Technical report, Los Alamos National Laboratory, Los Alamos, New Mexico.

[79] Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (1986). *Numerical Recipes in C++: The Art of Scientific Computing*. Cambridge University Press.

[80] Pusa, M. (2013). *Numerical Methods for Nuclear Fuel Burnup Calculations*. PhD thesis, VTT Technical Research Centre of Finland.

[81] Rhodes, J., Smith, K., and Lee, D. (2006). CASMO-5 Development and Applications. In *Proc. PHYSOR 2006*, Vancouver, Canada.

[82] Rhodes, R. (1995). *Dark sun : the making of the hydrogen bomb*. Simon & Schuster.

[83] Richards, S. D., Baker, C. M., Bird, A. J., Cowan, P., Davies, N., Dobson, G. P., Fry, T. C., Kyrieleis, A., and Smith, P. N. (2015). MONK and MCBEND: Current status and recent developments. *Annals of Nuclear Energy*, 82:63–73.

[84] Rief, H., Gelbard, E., Schaefer, R., and Smith, K. (1986). Review of Monte Carlo Techniques for Analyzing Reactor Perturbations. *Nuclear Science and Engineering*, 92:289–297.

[85] Robbins, H. and Monro, S. (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400–407.

[86] Romano, P. K., Horelik, N. E., Herman, B. R., Nelson, A. G., Forget, B., and Smith, K. (2015). OpenMC: A state-of-the-art Monte Carlo code for research and development. *Annals of Nuclear Energy*, 82:90–97.

[87] Rosenbluth, M. N. and Rosenbluth, A. W. (1955). Monte carlo calculation of the average extension of molecular chains. *The Journal of Chemical Physics*, 23(2):356–359.

[88] Senecal, J. P. and Ji, W. (2017). Approaches for mitigating over-solving in multiphysics simulations. *International Journal for Numerical Methods in Engineering*, 112(6):503–528.

[89] Shen, Q., Adamowicz, N., and Kochunas, B. (2019). Relationship Between Relaxation and Partial Convergence of Nonlinear Diffusion Acceleration for Problems with Feedback. In *Proc. M&C 2019*, Portland, Oregon.

[90] Sjenitzer, B. (2013). *The Dynamic Monte Carlo Method for Transient Analysis of Nuclear Reactors*. PhD thesis, Delft University of Technology.

[91] Soppera, N., Bossant, M., and Dupont, E. (2014). JANIS 4: An improved version of the NEA Java-based nuclear data information system. *Nuclear Data Sheets*, 120:294–296.

[92] Sutton, T. M. (2013). Anomalous Behavior of Monte Carlo Uncertainties near Reflecting Boundaries. In *Proc. SNA + MC 2013 - Joint International Conference on Supercomputing in Nuclear Applications + Monte Carlo*, Paris, France.

[93] Sutton, T. M. and Mittal, A. (2017). Neutron clustering in Monte Carlo iterated-source calculations. *Nuclear Engineering and Technology*, 49(6):1211–1218.

[94] Tuominen, R. (2015). *Coupling Serpent and OpenFOAM for neutronics - CFD multi-physics calculations*. Master's thesis, Aalto University.

[95] Turner, J. A., Clarno, K., Sieger, M., Bartlett, R., Collins, B., Pawlowski, R., Schmidt, R., and Summers, R. (2016). The Virtual Environment for Reactor Applications (VERA): Design and architecture. *Journal of Computational Physics*, 326:544–568.

[96] Ueki, T. (2008). On-the-fly Judgments of Monte Carlo Fission Source Convergence. *Transactions of the American Nuclear Society*, 11:512–514.

[97] Valtavirta, V. and Leppänen, J. (2018). New stochastic substep based burnup scheme for Serpent 2. In *Proc. PHYSOR 2018*, Cancún, Mexico.

[98] Wang, K., Li, Z., She, D., Liang, J., Xu, Q., Qiu, Y., Yu, J., Sun, J., Fan, X., and Yu, G. (2015). RMC - A Monte Carlo code for reactor core analysis. *Annals of Nuclear Energy*, 82:121–129.

[99] Wieselquist, W. (2015). The SCALE 6.2 ORIGEN API for High Performance Depletion. In *Proc. ANS M&C2015 - Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method*, Knoxville, Tennessee.

[100] Xu, X. G., Liu, T., Su, L., Du, X., Riblett, M., Ji, W., Gu, D., Carothers, C. D., Shephard, M. S., Brown, F. B., Kalra, M. K., and Liu, B. (2015). ARCHER, a new Monte Carlo software tool for emerging heterogeneous computing environments. *Annals of Nuclear Energy*, 82:2–9.

[101] Zoia, A., Brun, E., and Malvagi, F. (2014a). Alpha eigenvalue calculations with Tripoli-4®. *Annals of Nuclear Energy*, 63:276–284.

[102] Zoia, A., Dumonteil, E., Mazzolo, A., De Mulatier, C., and Rosso, A. (2014b). Clustering of branching Brownian motions in confined geometries. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 90(4):042118.